PERFORMANCE EVALUATION OF ECONOMIC DENIAL OF SUSTAINABILITY (EDOS) ATTACK MITIGATION TECHNIQUES

BY

Omar Ali Maraqa

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

December, 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Omar Ali Maraqa** under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS.**

Dr. Marwan Abu-Amara
(Advisor)

Dr. Ahmad Almulhem
Department Chairman

Dr. Ashraf Mahmoud
(Member)

Dr. Salam A. Zummo
Dean of Graduate Studies

Dr. Yahya Osais
(Member)

19/1/17
Date

Dedication

I dedicate this work to my parents, my brothers.

Thank you for supporting me along the way.

Without your praying and help, I could not have completed this work.

ACKNOWLEDGMENTS

All thanks and praises to ALLAH, the Almighty, for giving me the patience throughout this work, and the ability to complete it.

I would like to express my immense gratitude and grateful acknowledgment to my research supervisor, Dr. Marwan Abu-Amara, for his excellent guidance, constant encouragement, great efforts and support which helped me in completing different phases of this thesis. I would like also to thank my thesis committee, Dr. Ashraf Mahmoud, and Dr. Yahya Osais, for their valuable feedbacks which shape the final picture of this work.

I would like to thank King Fahd University of Petroleum and Mineral lecturers and staff for guiding me to accomplish my Master degree.

Finally, I would like to say thanks to my family and my friends for their continuous support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Full Name : Omar Ali Adel Maraqa

Thesis Title : Performance evaluation of Economic Denial of Sustainability (EDoS) Attack Mitigation Techniques.

Major Field : Computer Networks

Date of Degree : December 2016

Cloud computing technology is a result of urgent needs for low cost, high utilization, and efficient management of the available resources in the information technology industry. Many medium and large organizations are interested in cloud computing because of its benefits such as elasticity, pay per use, and other benefits that it provides. However, even with all of its great advantages, the security of cloud computing is still a major concern. Many new attacks have been developed especially for the cloud, and the Economic Denial of Sustainability (EDoS) attack is one of them. EDoS attack is considered one of the main security issues that prevents many organizations from migrating their services to cloud computing environment. EDoS targets the financial constraints of the cloud consumer who rents the resources from the cloud provider. A number of researchers proposed mitigation techniques that can reduce the effect of an EDoS attack.

In this work, we study the existing mitigation techniques that can mitigate the effect of the EDoS attack to come up with a comprehensive qualitative survey regarding such mitigation techniques. Moreover, we perform a thorough simulation validation for four of the proposed mitigation techniques that are considered having the most complete implementation details. The simulation validation is based on the use of a common

simulation platform, namely, CloudSim Simulator. In addition, we present a detailed quantitative simulation analysis for testing the suitability of these approaches in dealing with real cloud implementation conditions, such as different load balancing algorithms, different types of algorithms that identify the automated attackers, different probability distributions of request service time for cloud users (input traffic), the capability of these techniques in handling the cases when the cloud legitimate users and attackers belong to the same NAT-based network, and when cloud legitimate users generate a Flash over-Crowd (FC) traffic towards the cloud.

ملخص الرسالة

الاسم الكامل : عمر علي مرقة

عنوان الرسالة : تقييم أداء بعض الحلول التقنية التي تخفف من هجمات الحرمان الاقتصادي في أنظمة الحوسبه السحابيه

التخصص : هندسه الشبكات الحاسوبية

تاريخ الدرجة العلمية : ديسمبر, 2016

تم إنشاء أنظمة الحوسبة السحابية للحاجة الماسه لأنظمة تقوم بتوفير التكلفة حيث يتم استغلال الموارد الداخلية فيها وتنظيمها بشكل فعال. العديد من الشركات الكبرى والمتوسطة مهتمةٌ حالياً بأنظمة الحوسبة السحابية نظراً لتوفر الموارد الداخليه فيها بشكل مستمر ولوجود نظام الدفع حسب الاستخدام ولوجود العديد من الخدمات الاخرى التي توفرها هذه الانظمة. وعلى الرغم من كل هذه الميزات في أنظمة الحوسبة السحابية إلا انّ نظام الحماية من الهجمات في أنظمة الحوسبة السحابية ما زال يحتاج إلى الكثير من التطوير والتحسين. تم تصميم العديد من الهجمات خصيصاً لأنظمة الحوسبة السحابية وتعد هجمات الحرمان الاقتصادي إحدى هذه الهجمات. هجمات الحرمان الاقتصادي تعد إحدى أهم الاسباب التي تمنع العديد من الشركات من تبني واستخدام أنظمة الحوسبة السحابية; نظراً لانّ هذه الهجمات تزيد من الفاتورة التي تضطر الشركة لدفعها الى شركة الحلول التقنية المزودة لخدمة الانظمة السحابية.

في هذه الرسالة العلمية تمت دراسة أغلب أنظمة الحماية ضد هجمات الحرمان الاقتصادي وذلك للخروج بتقرير كامل ومفصل ليتم نشره في ورقة بحثيه. كذلك تم القيام بدراسة مفصله لأربع أنظمة حماية ضد هجمات الحرمان الاقتصادي والتي كانت تحتوي على معلومات كافيه لتسمح لنا بتطبيقها بأقل نسبه خطأ ممكنه. تم بناء هذه الانظمة الأربعة باستخدام برنامج محاكاه واحد وهو (CloudSim) وذلك للخروج بنتائج متناسقة. أخيرا تم إخضاع هذه الانظمة الأربعة لعملية محاكاه مفصله ومدعومة بالارقام تحت عوامل مختلفه مثل اختبار هذه الانظمه تحت خوارزميات توزيع طلبات مختلفه كذلك اختبار هذه الانظمه تحت أنماط مختلفة من اختبارات الرسم تيورنج (Turing tests) بالإضافة إلى اختبار هذه الانظمة مع اختلاف نمط إرسال الطلبات باتجاه السحابة من طرف المستخدمين وفحص مقدرة هذه الأنظمه على تحديد شرعية المستخدمين الكائنين خلف جهاز توجيه ترجمة عنوان الشبكة (NAT), كذلك تم فحص مقدرة هذه الأنظمة على كشف ظاهرة (Flash over-Crowd).

CHAPTER 1

INTRODUCTION

Cloud computing is a technology model which makes a huge revolution in the computing environment. Cloud computing is a utility that provides services on demand. All services provided by the cloud are elastic and could be leased by business companies via either a thin client interface (web browser) or thick client interface (program interface) through the Internet. These services are based on a model called “pay per use” model, which allows the cloud service consumers to request resources on demand and pay only for their usage. The cloud computing services can be categorized based on the type of resources provided by the cloud. These categories include the Infrastructure offered as a service (IaaS), Platform offered as a service (PaaS), and Software offered as a service (SaaS). There is another classification of cloud computing, which depends on the location of the cloud resources. This classification is divided into public cloud, private cloud, hybrid cloud, and community cloud. In addition, there are three main contributors in any cloud system, namely, the cloud service provider, the cloud service consumer and the cloud service customer [1].

Many new attacks have been developed especially for the cloud, and EDoS attack is one of them [2]. There are two types of mitigation schemes for defending EDoS attacks, namely, proactive scheme and reactive scheme [2].

1.1 Background and Terminology

In this section we will explain briefly the cloud characteristics and classification models as well as the definition of Denial of Service attack (DoS), Distributed Denial of service attack (DDoS), and Economic Denial of Sustainability attack (EDoS).

1.1.1 Cloud Essential Characteristics

According to the National Institute of Standards and Technology (NIST) there are six main characteristics that exist in any cloud computing environment [1, 3].

1. **Rapid elasticity:** It is the ability to scale the cloud resources up and down as needed. From the consumer point of view, the cloud appears to be infinite, and the consumer can utilize as little or as much computing power according to his demands.
2. **Measured service:** This indicates that all the aspects of the cloud service are monitored and controlled by the cloud service provider. This is crucial for resource optimization, billing, capacity planning, access control, and other tasks.
3. **Service level agreement (SLA):** It is a contract between the cloud service provider and the cloud service consumer, where the consumer specifies his requirements and the provider shows his commitment to them. Usually, SLA consists of items such as cloud security, cloud privacy, cloud servers uptime, and backup procedures.
4. **On-Demand self-service:** This aspect means that the cloud service consumer can use cloud resources as needed without any human interaction with the cloud service provider.

5. **Resource pooling:** This aspect means that the cloud sources (systems, applications or data) which are hosted in the same physical hardware can be rented to multiple consumers using a multi-tenant model.
6. **Broad network access:** This aspect means that all cloud service customers can access cloud resources through their heterogeneous thick or thin client platforms, such as workstations, laptops, tablets and mobile phones, and all the infrastructure needed for this is available in the cloud solution.

1.1.2 Cloud Contributors Models

There are three main parties in this model, namely, the cloud service provider, the cloud service consumer and the cloud service customer [3].

1. **The cloud service provider:** Represents the cloud company which delivers the service to the consumer.
2. **The cloud service consumer:** Represents one or more organization which actually uses the service.
3. **The cloud service customer:** Represents the employees or the clients of the consumer.

1.1.3 Cloud Delivery Models

According to NIST there are three main delivery models in the cloud [1, 3].

1. **Software as a Service (SaaS):** In this model the cloud consumer uses an application hosted in the cloud but without any means of controlling the underlying infrastructure of the cloud, which includes the operating systems, servers, storage, or network.

2. **Platform as a Service (PaaS):** In this model the cloud consumers have full control over their deployed applications and also sometimes over the hosting environment configuration settings, but still do not have the ability to manage or control the underlying infrastructure of the cloud.
3. **Infrastructure as a Service (IaaS):** In this model the cloud consumers have the ability to manage or control operating systems, servers, processing power, storage, or network components such as load balancers and firewalls.

1.1.4 Cloud Deployment Models

NIST defines four deployment models in the cloud [1, 3].

1. **Private cloud:** In this type the cloud resources are exclusively used by a single organization. As such, the resources may be operated and managed by the same organization, a cloud provider, or with some cooperation between them. The cloud resources may exist on or off the organization buildings.
2. **Community cloud:** In this type the cloud resources are exclusively used by multiple organizations that share the same interests of security requirements, policy, or common missions. As such, the resources may be operated and managed by one or more of these organizations, a cloud provider, or with some cooperation between them. The cloud resources may exist on or off the organizations buildings.
3. **Public cloud:** In this type the cloud resources are openly used by organization employees or by organization customers where the resources are usually owned and managed by the cloud provider. Also, the cloud resources exist in the cloud provider side.

4. **Hybrid cloud:** This type is a combination of the private cloud and public cloud. Usually an organization deploys this model to outsource non critical information to some public cloud provider, in the same time the organization deploys a private cloud for their critical business information.

1.1.5 Denial of Service (DoS) attack

As shown in Figure 1.1, a DoS attack is defined as an effort of one machine (attacker) to make some network or server unavailable to its clients or to severely degrade the quality of service in an unexpected manner [4].

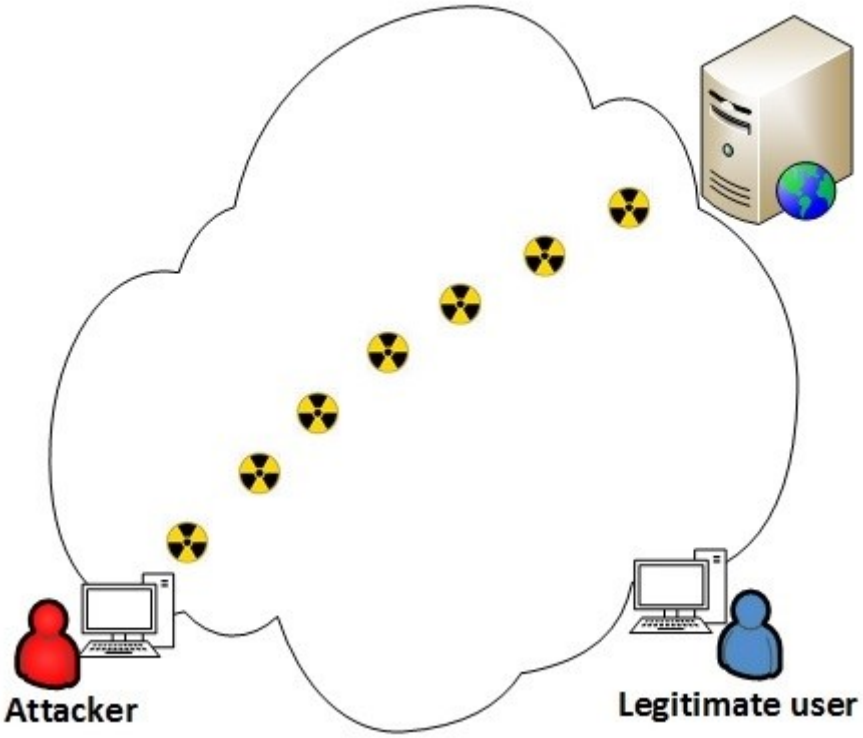


Figure 1.1: Representation of DoS attack.

1.1.6 Distributed Denial of Service (DDoS) attack

As shown in Figure 1.2, a DDoS attack represents the efforts of large number of machines to make some network or server unavailable to its clients or to severely degrade the quality of service in an unexpected manner [4].

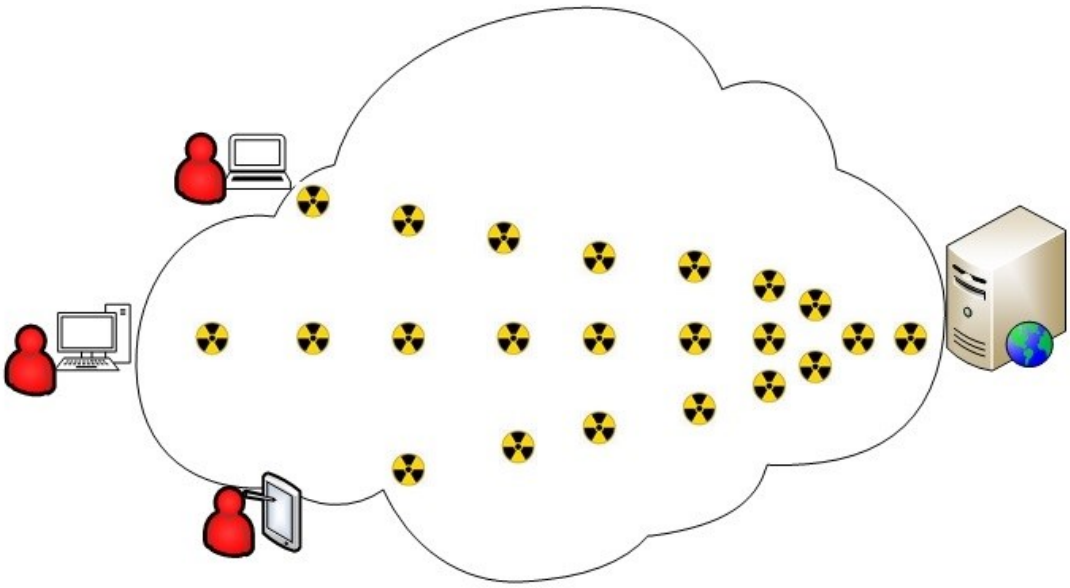


Figure 1.2: Representation of DDoS attack.

1.1.7 Economic Denial of Sustainability (EDoS) attack

The cloud service consumer signs an SLA according to a "pay-per-use" model with the provider. So, an organization is billed based on its cloud resources usage. An EDoS attack targets the cloud environment to cause an economic loss to the cloud consumer, which can in turn severely impact the provider financially. During an EDoS attack, the cloud resources of the consumer will expand in order to handle the requests of the attack due to the elasticity property of the cloud. Thus, the consumer needs to pay for all the cloud resources that have been allocated because of the attack. All these aspects are shown in Figure 1.3.

Figure 1.3: Representation of EDoS attack.

1.1.8 EDoS mitigation schemes

In general, there are two types of basic mitigation schemes for defending against EDoS attacks, namely, reactive scheme and proactive scheme [2]. Reactive mitigation schemes often proceed in three phases. In the first phase, distributed monitoring components try to detect the on-going EDoS attack. Once an attack is detected, the detector triggers the second phase that aims to locate the source of the attack. In the third phase, countermeasures are deployed to reduce the effect of the on-going attack. On the other hand, the proactive mitigation schemes intend to reduce the possibility of successful EDoS attacks by taking appropriate provisions prior to such attacks.

1.2 Problem Statement

EDoS attack is considered one of the main security concerns that have hindered the migration of many organizations from adopting the cloud technology. This is because an EDoS attack targets the financial constraints of the service consumer who rents the resources from the cloud provider. The EDoS attack exploits the elasticity feature of the cloud by forcing the cloud resources to scale up in order to accommodate all the service demand. As a consequence of "pay-per-use" model of the cloud, the service consumer will be charged as a result of the attackers activities.

In this work, we survey the existing mitigation techniques that attempt to mitigate the effect of the EDoS attack so as to come up with a comprehensive taxonomy survey regarding such mitigation techniques. Moreover, we perform a thorough simulation validation for four of the proposed EDoS mitigation techniques under one simulation platform (CloudSim simulator) so as to come up with consistent results for such techniques. Moreover, we present detailed quantitative simulation analysis for testing the suitability of these approaches in dealing with real cloud implementation conditions such as testing these techniques under different load balancing algorithms, different types of algorithms that identify the automated attackers, different probability distributions of request service time for cloud users (input traffic), the capability of these techniques in handling the cases when the cloud legitimate users and attackers belong to the same NAT-based network, and when cloud legitimate users generate a Flash over-Crowd (FC) traffic towards the cloud.

All the aforementioned performance analysis simulations that are considered in this work are currently missing from the literature even though such cases are expected to occur in a real cloud implementation.

1.3 Contributions

- Propose a comprehensive taxonomy of the existing mitigation techniques for EDoS attacks. We surveyed 16 mitigation techniques that can reduce the effect of EDoS attacks that are based on DoS or DDoS attacks.
- Perform thorough simulation validation for the approaches presented in [5] [6] [14] [21]. From the literature review, we have found that these four approaches represent the most detailed mitigation techniques for protecting cloud services against the EDoS attack. Specifically, these approaches provide proper description of the system architecture, and present the associated performance results. While validating the aforementioned solutions, we consider the following metrics: the utilization of the computing resources, and the cloud response time.
- Present a detailed quantitative simulation analysis for testing the suitability of these approaches in dealing with real cloud implementation conditions, such as, testing these techniques under different load balancing algorithms in order to pick one of the optimum solutions in this field, different types of algorithms that identify the automated attackers, different probability distributions of request service time for cloud users (input traffic), the capability of these techniques in handling the cases when the cloud legitimate users and attackers belongs to the same NAT-based network, and when cloud legitimate users generate a Flash over-Crowd (FC).

1.4 Thesis organization

The rest of the thesis is organized as follows. In Chapter 2, we provide a comprehensive survey of the research found in the literature for addressing the EDoS attack. Next, the methodology of the four mitigation techniques under study are fully described in Chapter 3. In Chapter 4, the simulation setup under the CloudSim simulator and the analytical model of the mitigation techniques under study are discussed. In chapter 5, we present the simulation validation results of the mitigation techniques under study. In chapter 6, we present the performance simulation results and their analysis for the considered mitigation techniques while taking into account different cases that aim to study the effect of different real cloud implementation conditions. Finally, chapter 7 includes the conclusion and directions for future work.

CHAPTER 2

LITERATURE REVIEW

In general, there are two main types of EDoS attacks; the network layer EDoS attack and the application layer EDoS attack [2]. The former tries to saturate the bandwidth of the links in the infrastructure of the cloud, while the latter tries to overwhelm the resources of the cloud servers. So, in this section, we summarize the research work found in the literature that attempt to mitigate the network level EDoS attack, the application level EDoS attack, and the techniques that can mitigate both types. Note that we present the EDoS mitigation techniques found in the literature according to the date of publication.

Khor and Nakao [2] described a first of its kind approach dedicated to reduce the effect of the EDoS attack in the cloud environment, the approach is called Self-verifying Proof of Work (sPoW). SPoW is designed to mitigate the network level EDoS attack by transforming its traffic into a new form which can be filtered by basic packet pattern-matching. Also, this algorithm can mitigate the application level EDoS by forcing cloud users to compete for cloud resources by solving a “crypto puzzle”.

In this approach, after the client requests the server access, the server asks the client to solve a “crypto puzzle” to prove the client commitment for its resources. The server also utilizes this “crypto puzzle” to protect the channel between the client and itself. The crypto puzzle consists of both the encrypted version of the server channel details and the encryption key with K bits which represents the difficulty of the puzzle. The client then consumes its resources to discover the details of the server channel and submit a connection

request towards the server through this secure channel. The connection request includes a random session key created by the client. Upon receiving this request, the server establishes a new permanent communication channel encrypted by the client's session key and the former server channel will be vanished. Accordingly, the authors prevent the network level EDoS attack from reaching the expensive cloud infrastructure by introducing the concept of the ephemeral server channel. Besides, the authors reduce the influence of the application level EDoS by introducing the "crypto puzzle".

This approach has several limitations such as a puzzle accumulation problem when there are a huge amount of fake puzzle requesters. Another limitations include asymmetric computation power for the cloud legitimate clients, and the puzzle generation cost at the server side. Finally, the authors did not provide an experimental work to highlight the performance of this solution [5].

Sqalli et al. [5] described a novel EDoS attack mitigation technique called EDoS-Shield. This technique is implemented to protect the cloud services from the application EDoS attacks by utilizing a Graphical Turing test. The EDoS-Shield protects also against the network EDoS attacks by using the Virtual firewall (VF).

The main idea behind the EDoS-Shield is to check whether the service requests are generated by legitimate users or come from bot machines. The architecture of the EDoS-Shield mitigation technique is shown in Figure 2.1, where the main two components of this approach are the VF and the Verifier Node (V-Node). The VF filters the incoming requests based on two lists; the blacklist and the whitelist. The V-Node is responsible for sending Graphical Turing tests such as CAPTCHA to the client and verifying the client response. If the client passes the CAPTCHA test then its IP address will be stored in the whitelist and

all subsequent requests from this IP address will be automatically directed towards the cloud resources without any further investigation. Otherwise, the client IP address will be stored in the blacklist and any subsequent requests from this IP address will be dropped [5].

This approach has some shortcomings manifested in its vulnerability to attacks that come from spoofed IP addresses which leads to the problems of false positive and false negative. The false positive appears when a blacklisted spoofed IP address is used by its original client. In this situation any traffic from this client will be dropped. On the other hand, the false negative appears when a whitelisted client changes its behavior to harm the cloud system by becoming an attacker. Another limitation is associated with requests that come from sources that lie behind a network address translation (NAT) router or behind a proxy. In this case the approach treats all the clients behind the NAT or proxy equally without distinguishing whether the clients are attacker or legitimate. In practice, it is quite possible to have both legitimate clients and bots behind the same NAT or proxy.

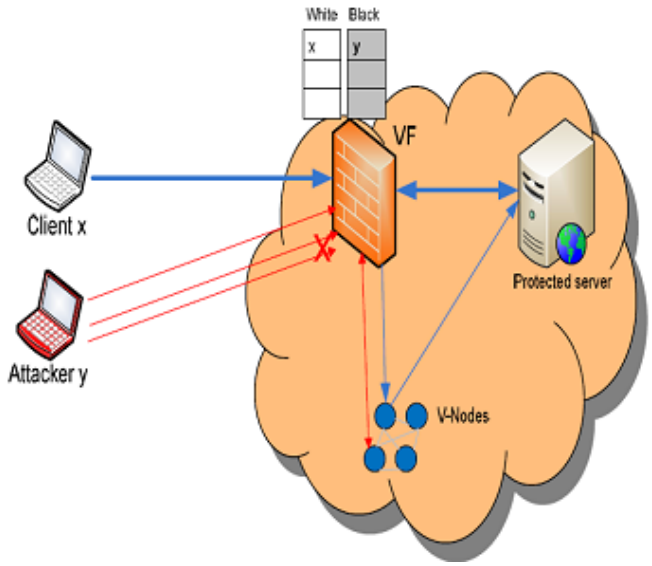


Figure 2.1: The EDoS-Shield architecture [5].

In [6], an enhanced version of the EDoS-Shield [5] that address the issue of IP spoofing is presented, where the authors append the Time to Live (TTL) value along with the IP address in the whitelist and the blacklist. In such a way, the authors can tell the malicious spoofed clients and legitimate clients apart. The limitation for this solution arises when the cloud sources lie behind a NAT or a proxy as the TTL value is not always accurate.

Kumar et al. [7] presented an in-cloud EDoS attack mitigation web service. The authors of this scheme designed their system to mitigate the network level EDoS as well as the application level EDoS using client cryptographic puzzles. This scheme includes three modules, namely, Proof of work technique, Packet filtering, and egress filtering. Only clients succeeding in solving the crypto-puzzle can gain the service access to the cloud resources. There are some shortcomings of this work. Firstly, the authors focus only on the parameters that make effective crypto-puzzle. Specifically the authors focus on how to make it easy to generate the puzzle by the server and difficult to solve by the client, while missing to describe the details of their architecture and the methodology of their algorithm. Secondly, this scheme is susceptible to puzzle accumulation attack at the server that generates the puzzles.

Kumar et al. [8] presented an EDoS mitigation technique in the cloud called In-Cloud Scrubber Service. According to the authors, this solution can mitigate the effect of the network level EDoS attack as well as the application level EDoS attack using an efficient client-puzzle approach.

In this technique, the authors add an on-demand servers to the cloud for generating and verifying crypto-puzzles with the clients. They refer to this service as the web Scrubber

service. The Scrubber service switches between three modes; normal mode, suspected mode with low-rated EDoS attack, and suspected mode with high-rated EDoS attack. The modes are dependent on the actual service provider server load and bandwidth load. During the suspected mode any incoming requests will be directed towards the In-cloud Scrubber service for verification process, while during the normal mode any incoming request will be immediately served by the actual cloud servers. There are two issues with the purposed scheme. The first one is the puzzle accumulation problem, and the second one is the fact that the authors did not provide experimental work to highlight the performance of this solution.

Sandar and Shenai [9] proposed an EDoS mitigation technique similar to the one proposed by the authors of [5]. The author in this technique implement a puzzle server that generates and verifies a client cryptographic puzzle instead of the CAPTCHA that is used by the authors of [5]. This solution is also proposed to protect the cloud services from a specific type of DDoS attack, the HTTP and XML based DDoS attack, that leads to EDoS to the cloud service under attack. In addition to the proposed mitigation technique, the authors also made qualitative comparison between different DDoS and EDoS countermeasures. As this technique is similar in nature to that presented in [5], it suffers from the same shortcomings as in [5].

Masood et al. [10] proposed a cost effective mitigation technique for EDoS attack called EDoS Armor. This work concentrated on the EDoS attacks that target the E-commerce applications hosted in a cloud system. This technique is implemented to protect the cloud application services from network level EDoS and application level EDoS attacks. The EDoS Armor includes three main modules: challenge server, admission

control, and congestion control, as shown in Figure 2.2. The challenge server is implemented to deal with the flooding attack that comes towards the cloud server. The challenge server can generate and verify either cryptographic or image based challenges. The admission control model is implemented to mitigate the network level EDoS attack by utilizing a port hiding mechanism, in which the attacker cannot perform a network level EDoS without knowing the system valid port. Moreover, with the admission control model, the number of simultaneous clients that access the cloud server can be limited to match the available cloud resources. The congestion control model is implemented to mitigate the application level EDoS attack by monitoring and prioritizing clients according to their browsing behavior. Specifically, any client that is involved in an intensive search queries without going to the purchasing phase is considered a bad client, and in return this client will face high response time for their malicious behavior.

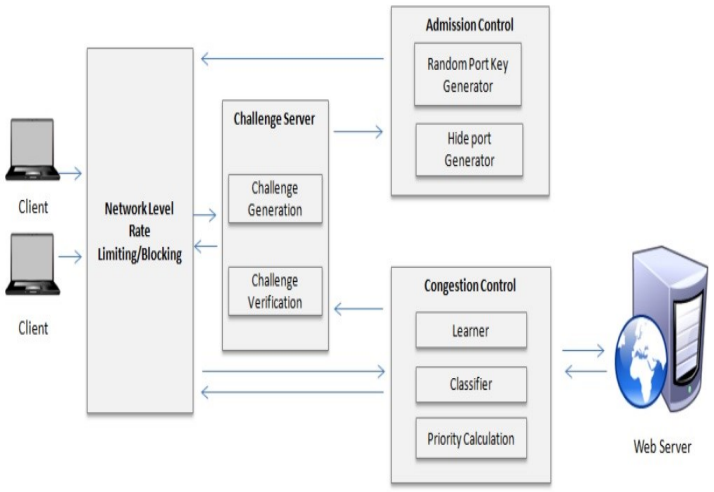


Figure 2.2: The EDoS Armor architecture [10].

The EDoS Armor has the following shortcomings. Firstly, this solution contradicts with the cloud scalability feature because the admission control model limits the number of simultaneous cloud users. Secondly, the average response time for good users is relatively high when compared with the results of [5].

Alosaimi and AlBegain [11] have presented a framework to mitigate the effect of the EDoS attack called DDoS Mitigation System (DDoS-MS). In this work the authors enhance the EDoS-Shield solution [5] by decreasing its end-to-end latency. Since it is an extension of the EDoS-Shied solution [5], the DDoS-MS can mitigate the network level and the application level EDoS attacks by testing only the first two packets of the client request.

The DDoS-MS consists of six main models: a Filtering Router, Green Nodes, a DNS server, a Virtual Firewall, a Client Puzzle Server, and a Verifier Node, as shown in Figure 2.3. The Virtual Firewall stores the IP addresses of the clients along with the Time to Live (TTL) value of the request in either a whitelist or a blacklist depending on the verification result. The Verifier Node uses a Graphical Turing test (GTT) for verifying the first packet of the request. The Client Puzzle Server tests the second packet of the request via a crypto puzzle to authenticate legitimate clients and avoid bots attacks. The authors implement the DNS server and the Green Nodes for hiding the location of the protected cloud server, while using the Filtering Router to forward only the packets that come from the Green Nodes to the protected server.

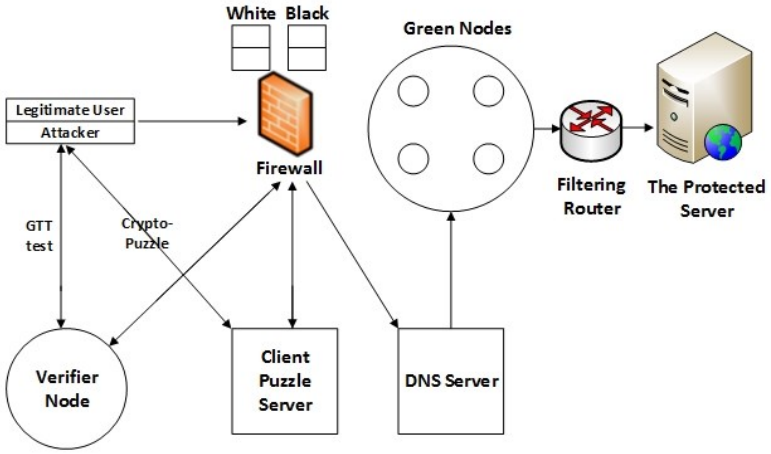


Figure 2.3: The DDoS-MS architecture [11].

The DDoS-MS has some shortcomings including the false negative problem, in which legitimate users may change their behavior to harm the cloud after the proposed algorithm verifies the users' first two packets [12]. Another shortcoming of the solution is that the authors did not provide an experimental work to examine the efficiency for this mitigation technique.

Alosaimi and AlBegain [12] presented an Enhanced DDoS-MS mitigation technique. The Enhanced DDoS-MS mitigation technique attempts to solve the false negative problem that exists in the DDoS-MS technique presented in [11]. Unlike the DDoS-MS [11], this solution only tests the first packet of each session.

The Enhanced DDoS-MS consists of five main models: the Reverse Proxy (RP) Server, Intrusion Prevention System (IPS), the Virtual Firewall (FV), the Verifier Node and the Client Puzzle server, as shown in Figure 2.4. There are four lists available in the FV: malicious, suspicious, black and white lists for the cloud service users. Those lists depend on the monitoring and verification results. The proposed solution has three verification layers. In the first layer, the verifier node verifies the first packet of the session to distinguish between the botnets and legitimate users using GTT. In the second layer, the IPS inspects packets flows to detect any malware components in these flows. If the IPS successfully detects a malware component then the IP address of the source will be stored in the malicious list. In the third layer, the RP server detects any suspicious user who tries to flood the system with requests. If one exists, then the Client Puzzle server sends a Crypto Puzzle to that user, forcing the user to consume its computational resources trying to solve this puzzle. In return, the user that generates huge amount of requests will be delayed. The

Enhanced DDoS-MS uses the Client Puzzle server as a reactive step to mitigate the effect of flooding attack by malicious users.

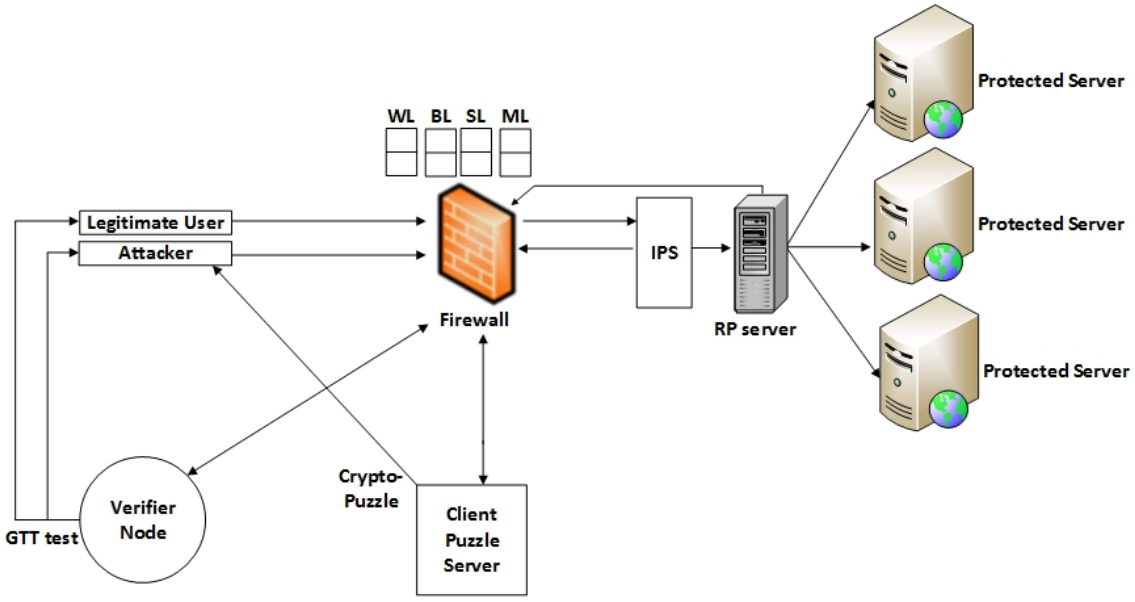


Figure 2.4: The Enhanced DDoS-MS architecture [12].

The Enhanced DDoS-MS has some shortcomings as it utilizes a huge amount of expensive systems in order to mitigate the EDoS attack. Also, the authors did not provide an experimental work to show the performance of the proposed technique.

M. Kumar and N. Roberts [13] presented a mitigation technique for the EDoS attack that is based on the public key infrastructure (PKI). This technique utilizes the Digital signature in such a way to provide mutual authentication between the server and the user. The proposed solution has two stages. At the first stage, the server issues a special certificate for each user that asks to access the cloud service. The certificate will be encrypted by the public key of a Certificate Authority (CA) and transmitted to the user. The user in turn transmits his original certificate encrypted with the public key of CA to the server. Then each side asks the CA to check the other side's certificate. Thereby, the proposed solution provides a two-way mutual authentication between both parties. After

that the server sends an encrypted unique password to the user. The password will be used in the future for data exchange. While at the second stage, the authors implement a Verifier Node to check whether the requests come from botnets or legitimate users using a specific hashing algorithm.

The proposed scheme suffers from the certificate accumulation problem. Moreover, the end-to-end delay is high in this algorithm since the mutual authentication phase requires the help of the CA. Furthers, the provided description of the experimental work is not clear.

Baig and Binbeshr [14] have described a scheme for mitigating and detecting the effect of EDoS attack on the cloud scalability feature. The proposed scheme depends on two factors to classify user requests as malicious or legitimate: the threshold and the duration, where the former refers to the maximum number of the requests beyond which the cloud scalability feature will be activated. While the latter refers to the length of time during which the scalability feature will be active.

There are four main components in the scheme: vFirewall, Job Scheduler, VM Observer, and Virtual Machine (VM) investigator, as shown in Figure 2.5. The vFirewall purpose is to analyze the incoming requests. If the request is received from a blacklisted user then the request will be sent to the VM investigator for further investigation, while if the source of the request appears in the white list then the source traffic will be directed to the cloud VM's. The Job Scheduler divides the requests between the individual VMs according to round robin scheduling algorithm. If the scheduling algorithm leads to overwhelm any of the cloud VM, then the VM Observer forwards the additional requests to the VM investigator for further analyses. When the VM investigator receives a request from either the vFirewall or the VM Observer, the VM investigator sends a Turing tests

toward the owner of that request. The purpose of the Turing test depends on the source of the received request. When the request comes from the vFirewall, the purpose of the test will be to check the legitimacy of the sender. On the other hand, when the request comes from the VM Observer, the purpose of the test will be to provide the additional users with a delayed access to the cloud service. The major limitation of this solution arises when the cloud sources lie behind a NAT or a proxy.

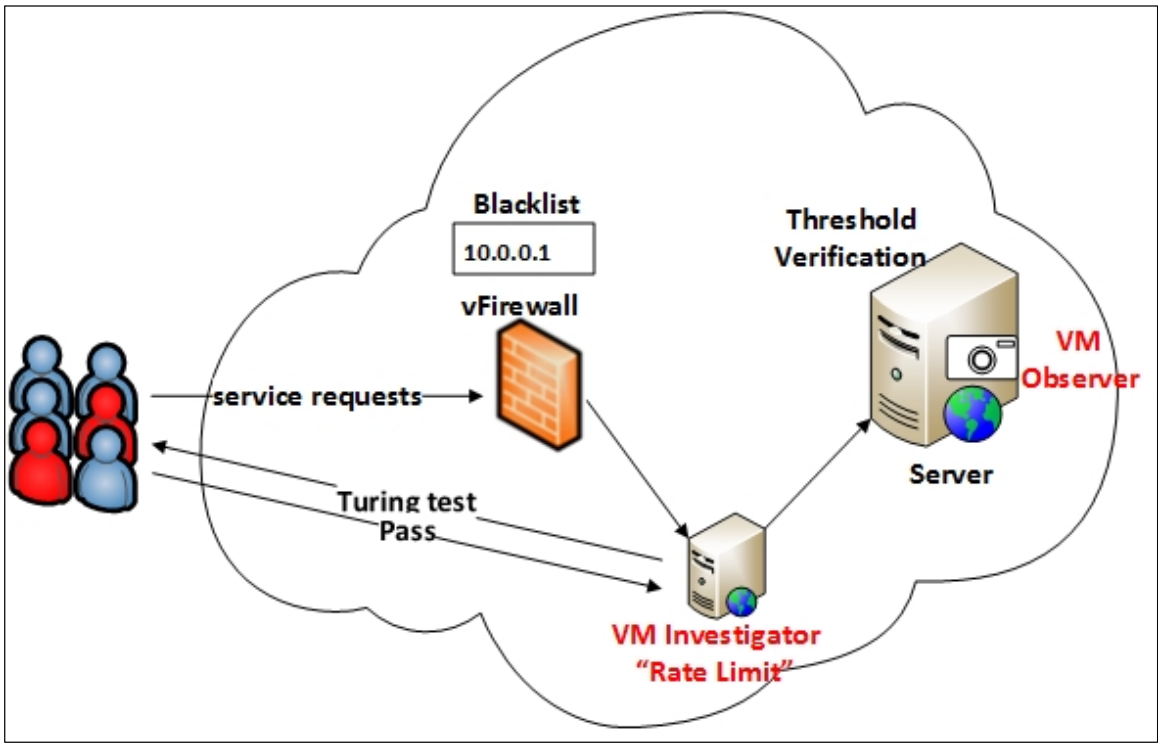


Figure 2.5: The Controlled Virtual Resources Access EDoS mitigation scheme Architecture [14].

Amazon in [15] provided a monitoring service for their cloud consumers to reduce the effect of the EDoS attack. The proposed service is called the Cloud Watch. Cloud Watch enables the cloud consumers to set an upper limit on their cloud platforms elasticity. A major shortcoming for this solution is that defining such a limit results in the loss of the scalability feature of the cloud. Also, the cloud consumers will still be charged according

to the pre-defined threshold of the cloud platforms elasticity because of serving the attack traffic. Moreover, this scheme is exposed to spoofed IP EDoS attacks [16, 17].

Anusha K. et al. [18] proposed a technique for detecting EDoS attacks based on the Time Spent on a Web Page (TSP) which represents the duration spent on viewing a website. A massive quantity of very few TSP values indicates a botnet targeting the web page. The average TSP value resulting from the attack payload is different from the mean TSP of a website. The TSP deviation from the mean value can be calculated in terms of Mean Absolute Deviation (MAD). A MAD plot method and foot step graph method; plot the deviation and the TSP's respectively to identify the various types of the attack traffic. However, the proposed technique requires human intervention to monitor and interpret the plots.

Saini and Somani [19] proposed a novel technique to decrease the effect of index page EDoS attack which targets the index page of a website. The proposed technique is called Index Page Attack Defender (IPA Defender). The index page attack is feasible because the website index page is freely available and accessible without any type of authentication.

The proposed scheme works as follows. Initially, the IPA Defender checks every request for the website index page. If the requester exceeds the page count threshold then the request and all subsequent requests will be dropped by the IPA Defender, and the requester IP address will be stored in the DROPLIST table for a specific amount of time. The proposed solution suffers from a number of issues. First, a poor description is provided about this solution. Second, the index page EDoS attack will be successful if the attackers design their attack to never exceed the page count threshold. Third, the proposed solution is susceptible to the false positive problem.

Al-Haidari et al. [20] presented an analytical model to study the influence of EDoS attack on the cloud service. The analytical model was validated using a discrete event simulation model. Also, the analytical model incorporated a queuing model that imitated the behavior of the cloud and considered some of the cost and performance metrics such as the resulted incurring cost of the attack, utilization, throughput, and end-to-end response time. The work result shows a considerable influence of EDoS attack on both the cost and the performance of the cloud service.

Baig et al. [21] modified their work in [14] by removing the VM observer and adding a Database (DB) next to the VM investigator to hold a copy of the blacklist users and some necessary variables for the operation. In addition, they removed the rate limit algorithm for the blacklisted users. Instead, they proposed a new algorithm called "limited access permission" to detect and mitigate the EDoS attack against the cloud service provider. Finally, they built a physical experimental setup in order to evaluate their proposed technique. The limitation for this solution arise when the cloud sources lie behind a NAT or a proxy.

Ficco and Rak [22] proposed a new EDoS attack mitigation technique that is based on the use of an intrusion Prevention system (IPS) and the adoption of the Service Level Agreement (SLA). The mitigation scheme splits end users into classes based on their IP addresses and the penalty cost of service unavailability that is defined by SLA. Then, the specific class that represents large number of bot machines and has the least penalty cost is prohibited from accessing the cloud resources. In such a case, the cloud provider accepts the idea of paying the service unavailability fees to the cloud consumer. In return, the cost of the cloud infrastructure remains acceptable.

The proposed scheme does not provide a method for detecting the presence of the EDoS attack, but rather it focuses on designing a new cloud architecture that can mitigate this kind of threats. This mitigation technique is lacking performance evaluation results in order to prove the feasibility of this scheme.

Table 2.1 summarizes the research found in the literature for addressing the EDoS attack, also it provides a modified version of the EDoS attack mitigation techniques summarization presented in [9] [16] [17]. From our literature study, we note that there are a couple of problems in the existing solutions. Some of the solutions are impractical due to the cost of the components which form the architecture of the solution. Also, some solutions are not adaptable to the dynamic environments as there are no learning mechanisms implemented in such schemes. Other schemes are susceptible itself to the DDoS attacks. Moreover, it is apparent that there is no concrete experimental study to help in choosing between all of these alternatives. Thus, in this work we will conduct a performance evaluation between the work presented in [5] [6] [14] [21] as they are considered having the most complete implementation details. Also, we come up with a generic criteria that can prove with measurable parameters the performance of these existing solutions. Such a performance evaluation platform can be easily adapted to test the performance of future solutions.

Table 2.1: EDoS Mitigation techniques summary [9] [16] [17].

Technique	**Metrics**			
	Methodology	Simulation Setup/Experimental work	Can mitigate Network and/ (or) Application EDoS?	Limitations
sPoW [2]	Packet matching algorithm and crypto puzzle	No/No	Yes/Yes	- Puzzle Accumulation problem. - A symmetric consumption power problem
EDoS-Shield[5]	Packet filtering and verification	Yes/No	Yes/Yes	- IP spoofing EDoS attack. - The case when attackers are behind the NAT or proxy is not addressed.
Enhanced EDoS-Shield[6]	Packet filtering and verification	Yes/No	Yes/Yes	- The case when attackers are behind the NAT or proxy is not addressed.
In-Cloud EDDoS Mitigation[7]	Packet filtering, egress filtering, Proof of work.	No/No	Yes/Yes	- Puzzle Accumulation problem. - Poor description about the three main modules.
In-Cloud Scrubber Service [8]	Crypto Puzzle	No/No	Yes/Yes	- Puzzle Accumulation problem.
Sandar and Shenai EDoS mitigation technique [9]	Crypto Puzzle	No/Yes	No/Yes	- Same drawbacks of [5].
EDoS Armor [10]	Admission control and Congestion control	Yes/Yes	Yes/Yes	- Contradict with the Cloud Scalability feature. - High response time for legitimate users.
DDoS-MS [11]	Packet filtering and verification	No/No	Yes/Yes	- False Negative problem.
Enhanced DDoS-MS [12]	Packet filtering and verification	No/No	Yes/Yes	- Expensive solution. - Cannot mitigate cloud internals attacks.
EDoS mitigation based on Digital signature[13]	Mutual authentication and Verification	Yes/No	No/Yes	- Certificate generation accumulation problem. - High end-to-end delay solution. - Not clear description of the experimental work.

Controlled Virtual Resources Access EDoS Mitigation[14]	Request Rate limit and Turing test	Yes/No	Yes/Yes	- The case when attackers are behind the NAT or proxy is not addressed.
Cloud Watch [15]	Traffic monitoring	Yes/Yes	No/Yes	- Contradict with the Cloud Scalability feature. - IP spoofing EDoS attack.
TSP EDoS mitigation technique [18]	Monitoring MAD and foot step plots	Yes/Yes	No/Yes	-requires human intervention to interpret MAD and foot step plots.
IPA – Defender[19]	Request Rate limit	No/Yes	No/ Yes	- A poor description is provided about the details of the solution. - Cannot mitigate page count threshold based EDoS attack. - False positive problem.
Controlled Access to Cloud Resources EDoS Mitigation[21]	Limited access permission and Turing test	No/Yes	Yes/Yes	- The case when attackers are behind the NAT or proxy is not addressed.
Ficco and Rak EDos Mitigation technique[22]	IPS and SLA	Yes/No	Yes/Yes	- There are no performance evaluation results in the paper in order to prove the feasibility of such scheme.

CHAPTER 3 THE METHODOLOGY OF THE EDOS MITIGATION TECHNIQUES UNDER STUDY

In this chapter, we present the main activities of the four EDoS mitigation techniques that were considered. Also, we discuss how each mitigation technique processes the cloud traffic in order to distinguish and mitigate the EDoS attack traffic from the normal traffic. This chapter is organized as follows, section 3.1 illustrates the methodology of the EDoS-Shield work. The methodology of the Enhanced EDoS-Shield is presented in section 3.2. Next, the methodology of Baig et al. mitigation techniques presented in [14] and [21] are discussed in section 3.3 and section 3.4, respectively.

3.1 EDoS-Shield

The main components of the EDoS-Shield mitigation technique are the virtual firewall (VF) and the verifier node (V-Node). The virtual firewall has two lists of IP addresses: whitelist and blacklist. The whitelist consists of those source IP addresses which are considered legitimate. All the requests that come from those sources are allowed to pass the firewall to the cloud servers. On the other hand, all the IP addresses that are contained in the blacklist are considered malicious, and hence all the traffic that comes from these IPs is blocked by the firewall [5].

When there is a request from a source, whose IP is not included in the firewall's lists, the request is forwarded to the V-Node. The V-Node sends a graphical Turing test to the request source. If the request has been issued by a human, the human will be able to pass the test by responding correctly to the test. Then, the V-Node will add the IP address

of the request source to the whitelist of the firewall and the request will be forwarded to the cloud server. Any following requests from this source will be allowed to pass the firewall. However, if the request has been generated by a machine such as bot, the machine will fail to solve the test. In this case, the V-Node adds the IP address of the request source to the blacklist of the firewall. Any following requests from this source will be blocked by the firewall [5].

3.2 Enhanced EDoS-Shield

Al-Haidari et al. [6] proposed a modified technique for their work in [5]. This technique attempts to detect EDoS attacks originating from IP spoofed addresses. The same architecture of the original EDoS-Shield is used, but with extra fields appended along with the sources IP addresses in the whitelist and the blacklist. The extra fields are the TTL values, a counter of unmatched TTL values in both the whitelist and the blacklist, and the attack start time field in the blacklist.

The TTL value is modified according to the verification phase done at the V-Node. The V-Node verifies client requests using Graphical Turing tests, such as CAPTCHA [23, 24]. If the source passes the test then the final value of the TTL will be placed in the whitelist along with the source IP address. If the source fails to respond to the test, the TTL value will be placed in the blacklist along with the source IP address [6].

The unmatched TTL counter field will be used to reduce the false positives requests. Instead of dropping packets because of not matching the TTL value, a verification phase will be performed at the V-Node as long as the “unmatched TTL” counter does not exceed

a given threshold. This will reduce the false positive results, since packets having different TTL values will still have a chance to verify their legitimacy at the V-Node [6].

The attack timestamp field in the blacklist is used to record the start time of the attack that is set to the time at which the source IP address is placed in the blacklist. The timestamp field will be utilized to make the verification phase at the V-Node more restricted during the attack. For example, if a packet arrives during the lifetime of the attack with a source IP address and TTL value that are present in the blacklist, it will be dropped without performing any further verification phase. On the other hand, if the packet arrives after the end of the attack period, then a verification phase will be performed since there is a probability that it is a non-spoofed request [6].

Figure 3.1 illustrates the actions at the vFirewall Node when receiving a packet. At the vFirewall, the packet will be forwarded directly to the destination only if its source IP address is found in the whitelist and its TTL value matches the TTL value that was stored in the whitelist. Otherwise, packets will be forwarded to V-Nodes for further investigation [6].

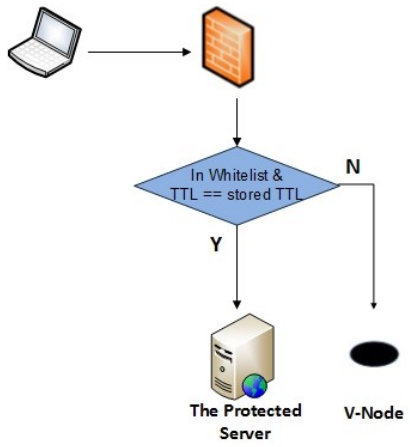


Figure 3.1: Main activity of vFirewall.

Figures [3.2-3.5] describe the actions at the V-Node when receiving a packet from the vFirewall. The V-node considers four cases based on the presence of the source IP address in the whitelist and/or blacklist. These cases are: the source IP address does not exist in the in the whitelist nor the blacklist, already present only in the whitelist, already present only in the blacklist, and present in both lists [6].

For the first case that is shown in Figure 3.2, where the source IP address is neither in the whitelist nor in the blacklist, the V-Node will perform a verification phase using Graphical Turing test. If the user passes the test, the user request will be forwarded to the cloud. Moreover, the user IP address along with the request TTL value will be placed in the whitelist and the unmatched counter will be initialized to zero. On the other hand, if the user fails to respond to the test, the user IP address along with the request TTL value will be placed in the blacklist, and the timestamp and unmatched counter will be initialized to the current time and zero, respectively [6].

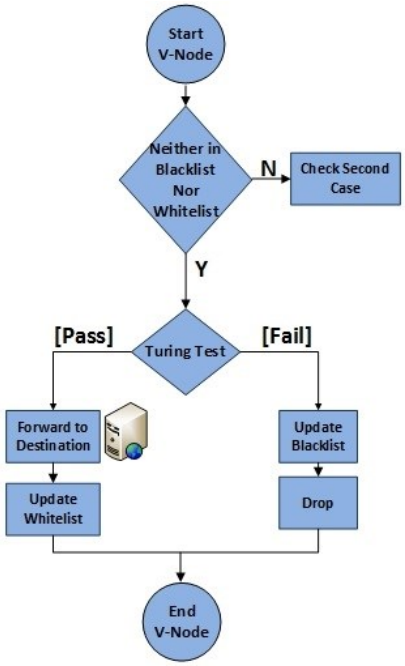


Figure 3.2: Main activity of V-Node when the source IP address is neither in the whitelist nor in the blacklist.

For the second case illustrated in Figure 3.3, where the source IP address appears only in the whitelist, the V-Node will perform a verification phase. If the user passes the test, the corresponding TTL value in the whitelist will be updated to the new value obtained from the last verified request. If the user fails to respond to the test, the unmatched TTL counter in the whitelist will be incremented and the source IP address will be added to the blacklist with its TTL value and timestamp [6].

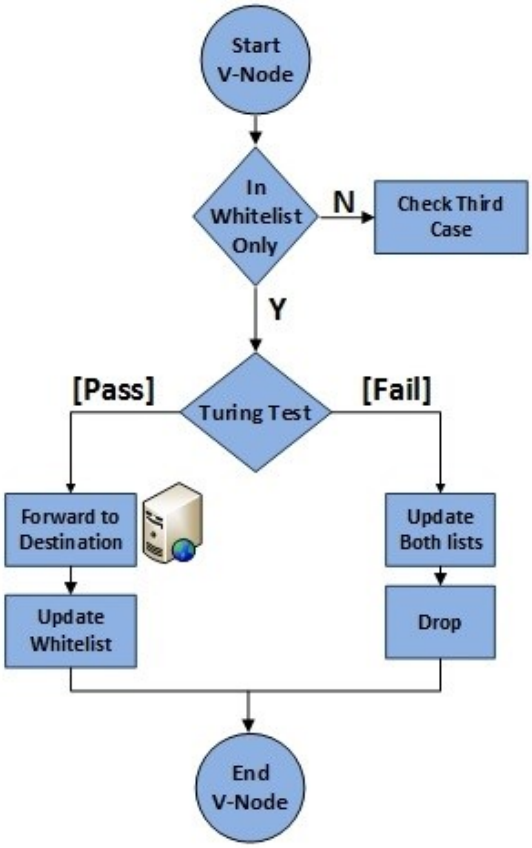


Figure 3.3: Main activity of V-Node when the source IP address appears only in the whitelist.

For the third case that is shown in Figure 3.4, where the source IP address appears only in the blacklist, the packet will be dropped when its TTL value matches the corresponding TTL value in the blacklist or when the unmatched TTL counter in the blacklist reaches the threshold during the attack's lifetime. Otherwise, the V-Node will perform the verification

phase. During the verification phase, if the user passes the test, the request will be forwarded to the destination and its source IP address will be placed in the whitelist along with the request TTL value. On the other hand, if the user fails to respond to the test, the packet will be dropped and the corresponding entry in the blacklist will be updated as follows. If the packet is received within the attack's lifetime, the unmatched TTL counter will be incremented. If it is received after the attack's lifetime elapses, the TTL, timestamp, and unmatched TTL counter fields in the blacklist will be set to the received packet TTL, current time, and zero, respectively [6].

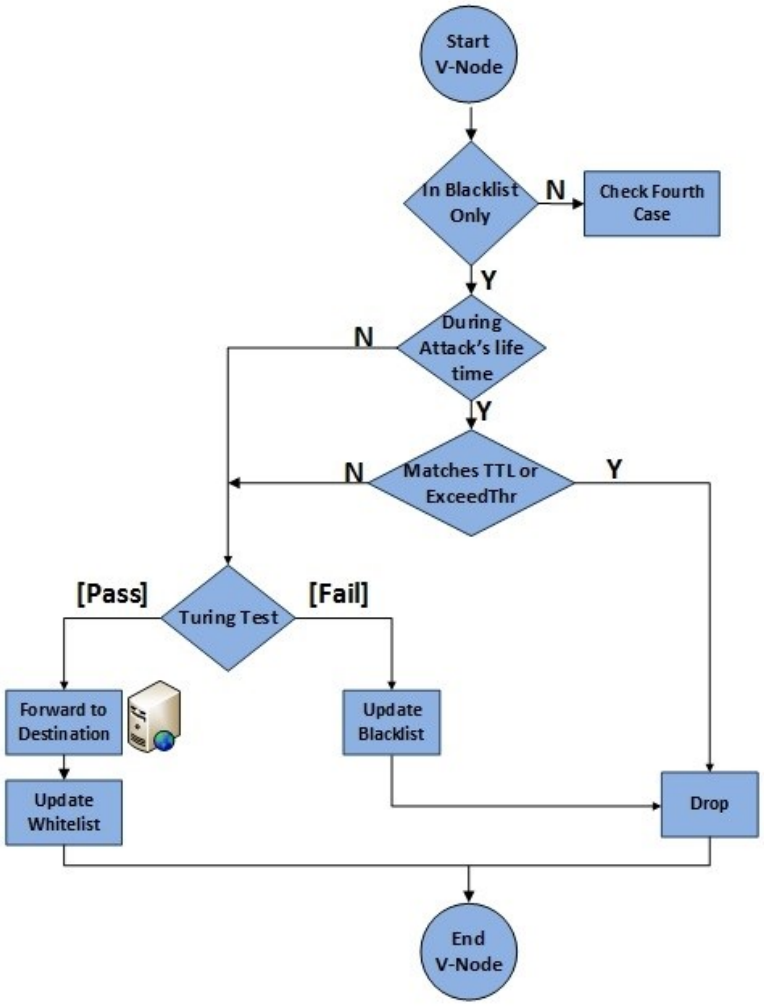


Figure 3.4: Main activity of V-Node when the source IP address appears only in the blacklist.

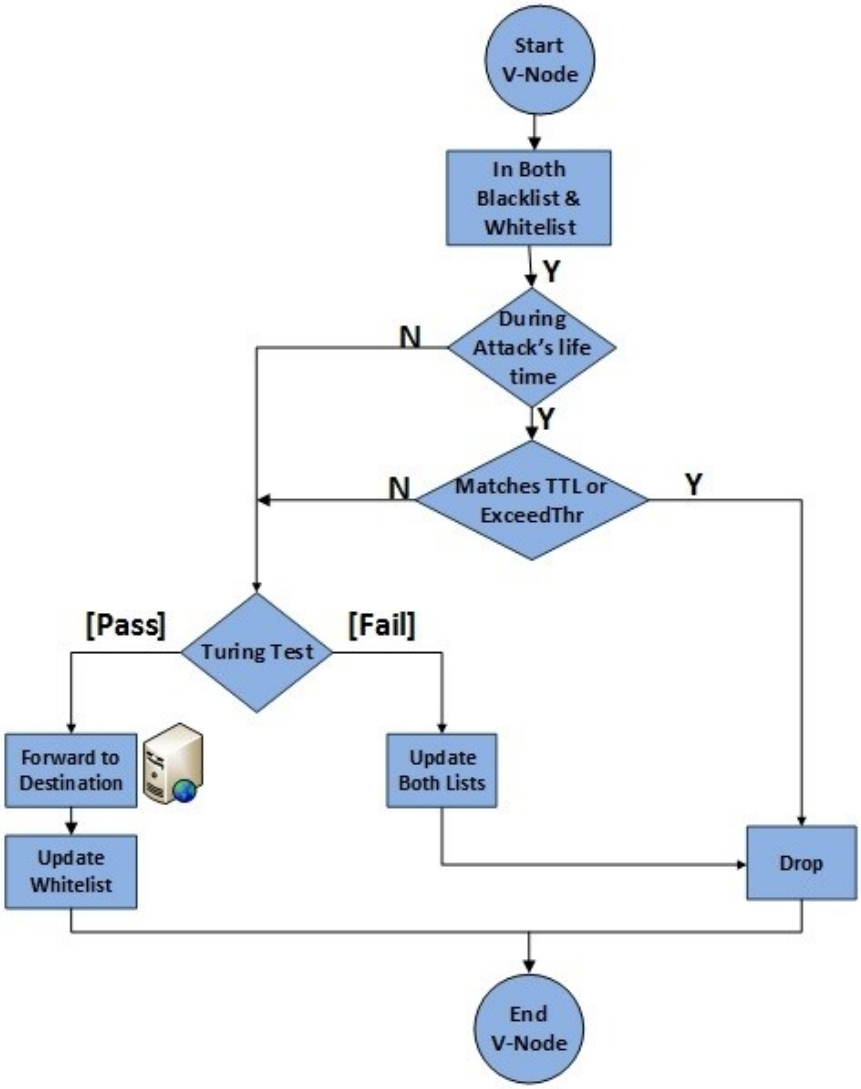


Figure 3.5: Main activity of V-Node when the source IP address appears in both lists.

For the fourth case depicted in Figure 3.5, where the source IP address appears in both lists. This means that the incoming traffic at the V-Node side might have some requests with spoofed IP addresses and others being legitimate requests. In such a case, the request will be dropped when its TTL value matches the stored TTL value in the blacklist of the same IP address, or when the unmatched TTL counter in the whitelist reaches the specified threshold within the attack's lifetime. Otherwise, the V-Node will perform a verification phase. If the user passes the test, the request will be forwarded to the destination and its corresponding entry in the whitelist will be updated by the new TTL value and by resetting

the unmatched TTL counter to zero. On the other hand, if the user fails to respond to the test, the packet will be dropped and the unmatched TTL counters in both the whitelist and the blacklist will be incremented if the packet is received within the attack's lifetime. Similarly, if the packet arrives after the attack's lifetime elapses, then the corresponding entry in the blacklist is updated so that the TTL, timestamp, and unmatched TTL counter fields, will hold the received packet TTL, current time, and zero, respectively [6].

3.3 Controlled Virtual Resources Access EDoS Mitigation

This method is proposed to detect the EDoS attacks that target the cloud service provider and to reduce the effects of these attacks based on a rate limit mechanism. The proposed method depends on two factors to classify user requests as malicious or legitimate; the threshold and the duration. The former refers to the upper limit of the user's requests, beyond which the cloud scalability feature will be activated. While the latter refers to the length of time in which the scalability feature will be active [14].

This approach is considered as a reactive scheme to mitigate the EDoS attack because it starts running when the cloud provider side receives requests that exceed the threshold parameter [14].

There are four main components in the scheme, namely, vFirewall, Job Scheduler, VM Observer, and VM investigator, as presented in Figure 3.6. The vFirewall analyses the incoming request to decide if the request comes from a blacklisted user. If so, then the request will be sent to the VM investigator for further investigation. Otherwise, the request will be directed to the cloud Virtual Machines (VM).

The Job Scheduler divides the requests between the individual VMs according to the Round Robin scheduling algorithm. If the scheduling algorithm leads to overwhelm any VM in the cloud, then the VM Observer forwards the additional requests to the VM investigator for further analyses. The VM investigator receives the user's requests from two sources: the vFirewall and the VM observer. Subsequently in both cases, the VM investigator sends a Turing test towards the owners of these requests. The purpose of the Turing test depends on the source of the received requests. When the requests come from the vFirewall, the purpose of the test will be to check the legitimacy of the senders. On the other hand, when the requests come from the VM Observer, the purpose of the test will be to provide the additional users with a delayed access to the cloud service [14].

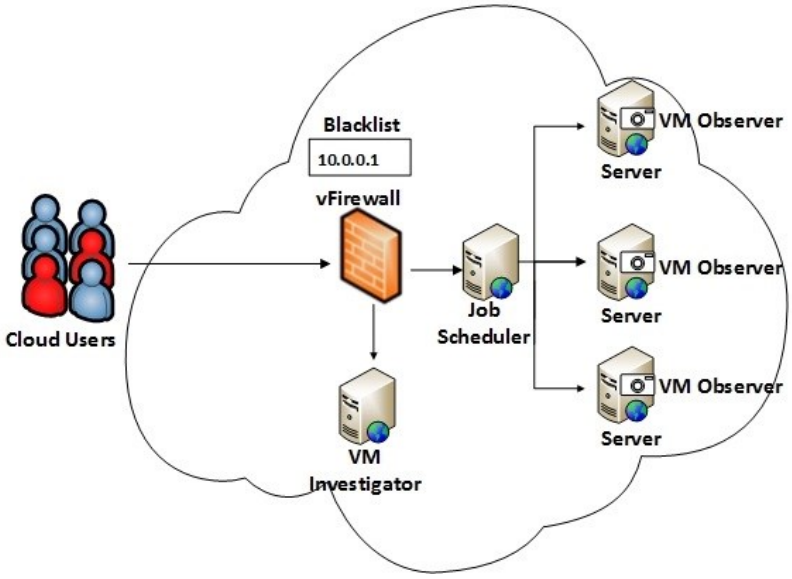


Figure 3.6: The Controlled Virtual Resources Access EDoS mitigation technique [14].

The Controlled Virtual Resources Access EDoS mitigation technique operates in three scenarios. The first scenario appears when the threshold parameter is not crossed, and when the service requests are received from non-blacklisted users. While the second scenario appears when the threshold parameter is crossed, and when the service requests are

received from non-blacklisted users. Finally, the third scenario appears when the threshold parameter is crossed, and when the service requests are received from a blacklisted user.

In the first scenario, the vFirewall forwards all the incoming requests to the VM directly, the VM observer checks the VM threshold parameter continuously, if the threshold parameter is crossed, the scheme will operate in the second scenario. Figure 3.7 describes the communication hierarchy of the first scenario [14].

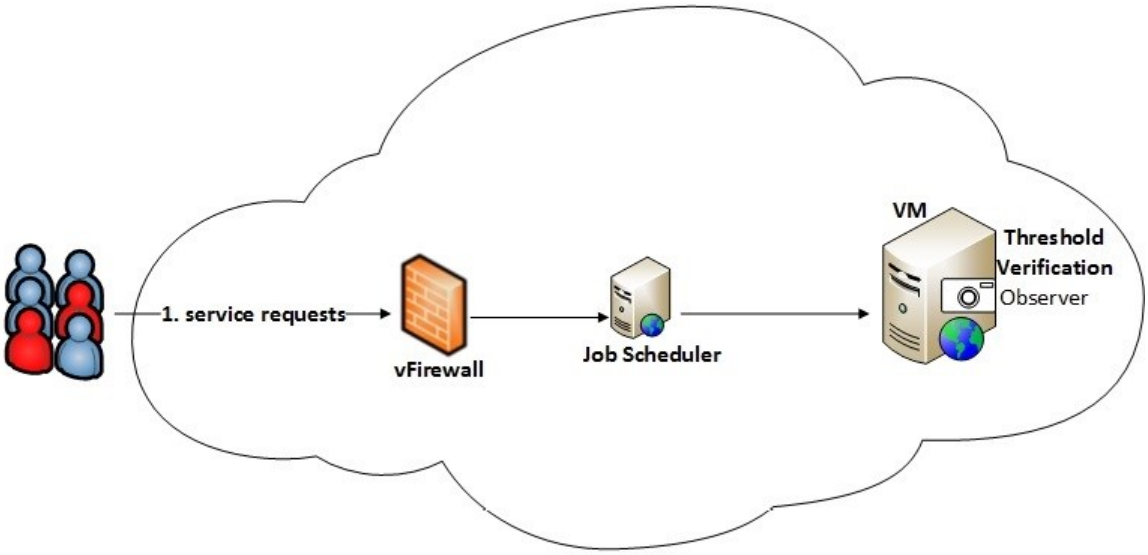


Figure 3.7: The communication hierarchy of the first scenario.

In the second scenario, the vFirewall forwards user's requests to the cloud resources. At this point, the VM is over-utilized so the VM Observer forwards all subsequent requests to the VM investigator. In return the VM investigator sends Turing test towards the user. The VM investigator initiates a User Trust Factor (UTF) parameter for each user (0<UTF<1). The VM investigator assigns a UTF of 0.5 to the new users, if the user successfully passes one Turing test the UTF for this user will be incremented 0.05. On the contrary, if the users fails the UTF will be decremented by 0.05. If a given user reaches a UTF of zero, the VM

investigator adds the user to the blacklist and all the subsequent requests from this user are dropped by the VM investigator. On the other hand, if the user reaches a UTF of 1, the VM investigator removes the user from blacklist and conveys this information to the vFirewall. For users that fail to solve the Turing test the VM investigator adds them to the black list then applies a rate limit algorithm on them. In such a case the VM investigator calculates a new parameter called "number of access to give" (w_{opt}) which refer to an upper bound of requests that the user can send in a limited time. The calculation of w_{opt} will be illustrated in chapter 4. The communication hierarchy of the second scenario is illustrated in Figure 3.8 [14].

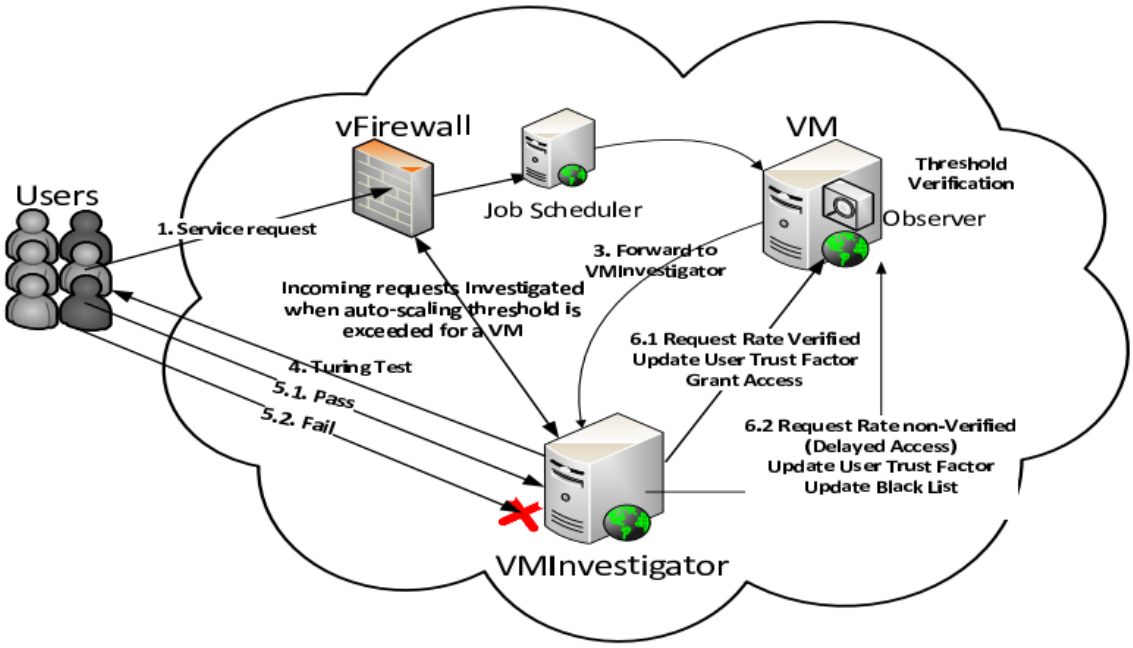


Figure 3.8: The communication hierarchy of the second scenario [14].

In the third scenario, the vFirewall forwards the blacklisted user's requests to the VM investigator, in which the scheme gives another chance for the blacklisted user to prove its legitimacy. If the user solves the Turing test, then the user will gain access to the cloud resources. On the other hand, if the user fails the test, the user will suffer from the rate limit algorithm. Figure 3.9 describes the communication hierarchy of the third scenario [14].

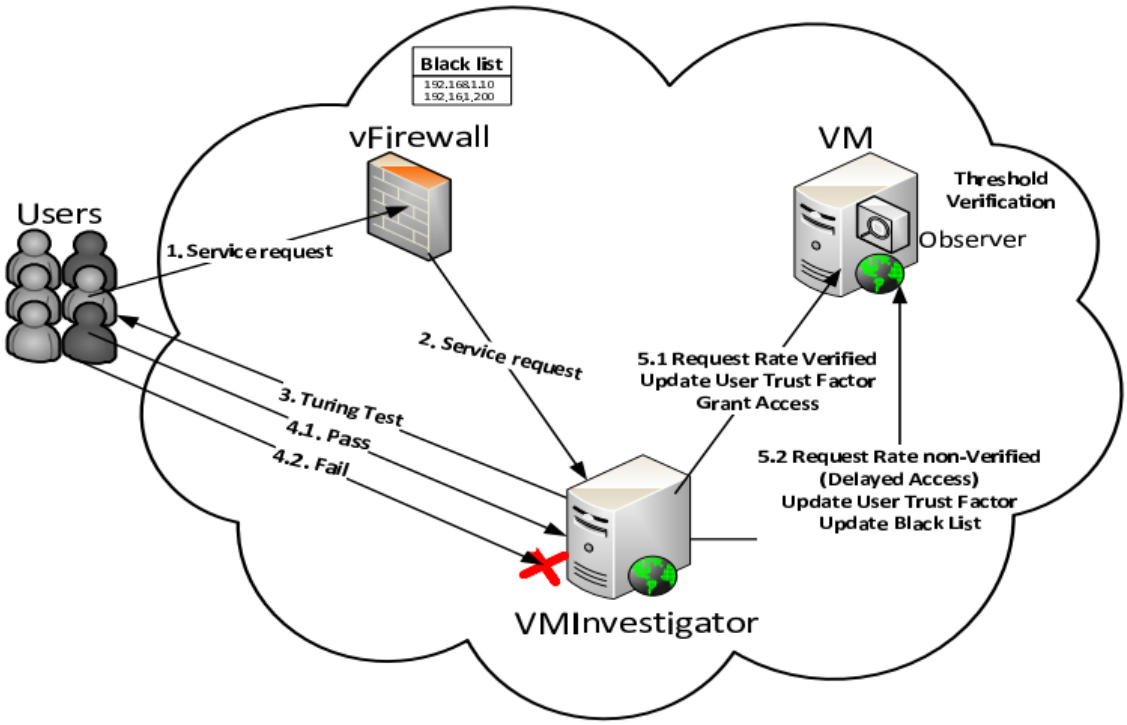


Figure 3.9: The communication hierarchy of the third scenario [14].

3.4 Controlled Access to Cloud Resources EDoS Mitigation

The proposed work [21] is a modified version of the work of Baig et al. [14]. In this technique, the authors add a new algorithm called “limited access permission” in the VM Investigator to detect and mitigate the EDoS attack. Also, the authors add a database (DB) that holds some necessary parameters for the operation of this technique.

The DB contains two tables, namely, the blacklisted table and the rate limit table. The blacklisted table has the IP address of the malicious users. On the other hand, the rate limit table stores five parameters, namely, the IP addresses of the cloud service legitimate and malicious users, the last activity timestamp which reflect the last seen activity of the user, the user requests count that records the number of requests that are made by a single user in one minute, the UTF where this parameter maintains a value between 0 and 1 that classifies the legitimacy of the cloud service users, and the count (CRPS) which hold the number of requests of a single cloud user in a single second.

Figure 3.10 shows the limited access permission algorithm that is deployed in the VM investigator. This algorithm depends on three main parameters: the UTF, the Concurrent requests per second (CRPS), and the Random Check (RC).

The UTF classifies users into three levels: bad, average and good. If the cloud user responds correctly to the Turing test then the UTF is incremented by 0.01, on the other hand if the user fails to respond to the test the UTF is decremented by 0.02.

The CRPS holds a value that defines an upper limit of the number of requests that one user can send in one second. The network administrator can adjust the value of the CRPS based on the history of the cloud service.

The RC parameter is used to counter smart attacker who can figure out the accurate CRPS and send requests less than the CRPS threshold. RC is an interval between one and the total requests per minutes (TRPM), where TRPM equals to 60*CRPS. This interval is divided into equal subintervals with the number of subintervals being equal to the CRPS value. Then, one value is chosen from each subinterval randomly. The chosen values are called RC_values_count. Finally, the RC_values_count is compared with the requests count parameter. If there is a match then a Turing test is sent to the end user, as shown in the left branch of Figure 3.10. For example, if the CRPS equals 2, then the RC interval is between [1, 2*60]. The interval is divided into two subintervals because the CRPS equals 2. The subintervals are [1, 60] and [61,120]. One number is chosen randomly from each subinterval. When a request arrives at the VM Investigator, the VM Investigator checks the requests count parameter in the DB. If it is a match, then the VM Investigator sends a Turing test to the user. On the other hand, if the requests count parameter does not match the RC_values_count, the decision of sending a Turing test toward the user is being made according to the UTF value of the user, as depicted in Figure 3.10.

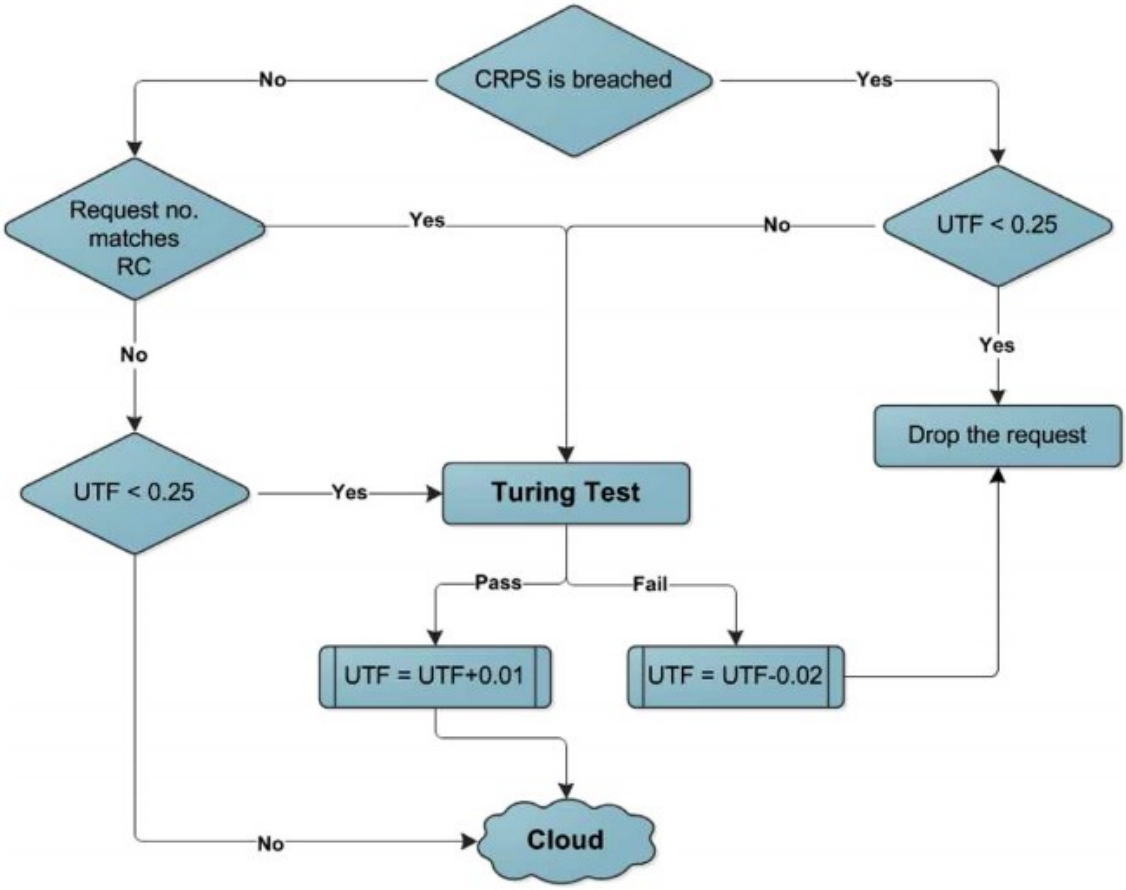


Figure 3.10: The VM Investigator flow chart [21].

CHAPTER 4 SIMULATION SETUP OF EDOS MITIGATION TECHNIQUES UNDER STUDY

In this chapter, we discuss the simulator implementation used to build the four mitigation techniques under study. In addition, we present the analytical and the simulation models for each considered mitigation technique. These models are used in chapter 5 for the validation purpose of each technique.

4.1 CloudSim Simulator

The CloudSim simulation tool [25] is used to implement the four EDoS mitigation techniques under study since this simulator has been widely used by the research community. Moreover, CloudSim enables researchers to focus on specific system design issues rather than being concerned with the low level details related to Cloud-based infrastructures and services [26].

The CloudSim simulator is implemented using the Java programing language. It is used to simulate different scenarios of cloud computing infrastructure. It provides different classes that describe cloud users (cloudlet), load balancer (broker), datacenter, virtual machine instances, random generators, storage elements, and management policies such as cloudlet scheduling policies and VM allocation policies [26].

The CloudSim frame work is designed as a multilayered software. It contains two main layers; the user code layer and the CloudSim layer [26]. The user code layer is concerned with the cloud basic units such as load balancing scheduling policies, virtual machines and their specifications, input traffic demand, and the number of cloud users. On the other hand,

the CloudSim layer is concerned with supporting the modeling and simulation of the virtualized cloud based datacenter that includes bandwidth, storage, memory, and virtual machines interfaces. Figure 4.1 illustrates the CloudSim architecture of these two layers.

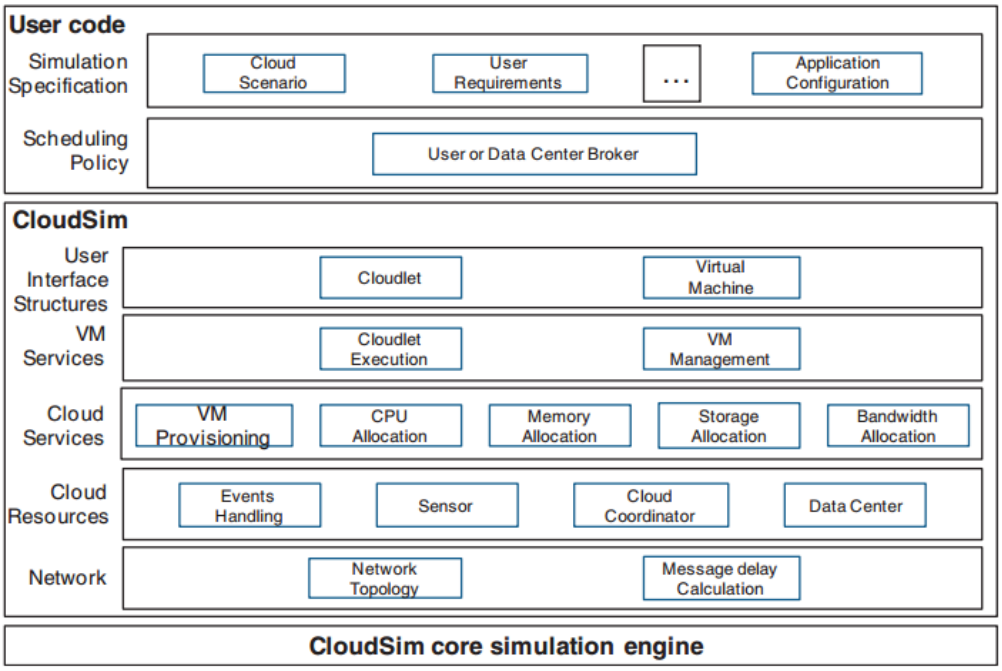


Figure 4.1: The CloudSim simulator Architecture [26].

In the CloudSim simulator, the average cloud response time is measured by collecting two main parameters for all user's requests (N) served by the cloud; the request arrival time (AT) and the request departure time (DT). The average cloud response time is calculated using the following equation:

$$RT_{\text{Avg}} = \frac{1}{N} \sum_{\text{i}=1}^{N} (DT_{\text{i}} - AT_{\text{i}}) \tag{4.1}$$

Also, the average CPU utilization of all cloud instances is measured by collecting the total server processing time and the total time of the simulation as shown in the following equation:

$$U_{\text{Avg}} = \frac{1}{S} \sum_{\text{i}=1}^{S} \frac{ProcessingTime_{\text{i}}}{RunningTime_{\text{i}}} \tag{4.2}$$

Where the $RunningTime_{\text{i}} = FinishingTime_{\text{i}} - InitiationTime_{\text{i}}$.

4.2 EDoS-Shield Analytical Model

This section presents the analytical model that was used to build the EDoS-Shield mitigation technique [5]. Figure 4.2 shows the queuing network model that represents the proposed mitigation technique. The input to this model is an aggregate traffic from both legitimate users and attackers. Sqalli et al. [5] have assumed a Poisson distribution for characterizing the EDoS traffic, Since in the literature many authors used the assumption of Poisson distribution to characterize DDoS attacks [27, 28], and the flooding nature of DDoS attack is similar to the EDoS attack.

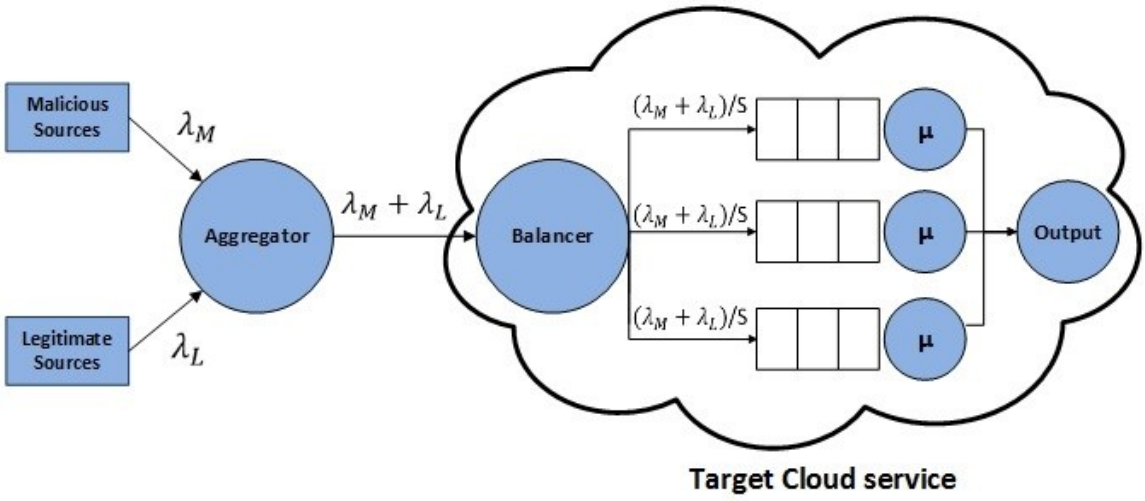


Figure 4.2: EDoS-Shield queueing model [5].

In order to evaluate the EDoS-Shield, we consider different performance metrics, namely, the cloud response time, computing resource utilization, and the number of allocated VMs.

In order to formulate the total cloud response time for the queueing model in Figure 4.2, the authors used the decomposition method which is discussed by chandy and Sauer [29]. Hence, the EDoS-Shield queuing model is broken up into three subsystems; vFirewall, vNode, and the cloud VMs. Then, the average delay is computed for each subsystem.

Finally, the total cloud response time is the summation of the average delay for each subsystem. Thus, the total response time is computed as follows:

$$Total\ response\ time = Average\ delay\ in\ vFirewall + \ average\ delay\ in\ links + \ average\ delay\ in\ VMs$$

Note that, because only few requests go through the vNode for verification purpose, the authors ignore the average delay incurred in the vNode.

Al-Haidari et al. [6] have modeled the links as M/D/1 queueing system with exponential arrival time and deterministic service time. The average delay that passes through any link provided by the following equation [30]:

$$link\ delay = \left(1 - \frac{\lambda_{\text{in}}}{\mu_{\text{link}}}\right) / (\mu_{\text{link}} - \ \lambda_{\text{in}}) \tag{4.3}$$

Where λ_{in} is the mean request arrival rate and $\mu_{\text{link}}$is the mean link request service rate.

The vFirewall and VMs can be modeled as a collection of parallel single queues, as described in [31]:

$$vFirewall/VM\ delay\ = \frac{S}{S_{\mu} - \lambda} \tag{4.4}$$

Where S is the total number of VMs, λ is the total arrival rate for the VMs, and μ is the service rate of a single VM.

Sqalli et al. [5] have also ignored the delay in the link between the end user and the vFirewall because they focused on the performance of the cloud that starts with the vFirewall.

Based on equations (4.3) and (4.4), we can formulate the total cloud response time, (RT), as follows:

$$RT = \frac{S_1}{S_1.\mu_1 - \lambda_1} + \left(1 - \frac{\lambda_1}{\mu_{\text{link2}}}\right)/(\mu_{\text{link2}} - \lambda_1) + \frac{S_2}{S_2.\mu_2 - \lambda_2} \quad (4.5)$$

Where μ_{link2} is the capacity of the link from the vFirewall to the cloud resources, S_1 is the number of the instances representing the vFirewalls, S_2 is the number of VMs representing the cloud resources, μ_1 is the processing rate of a vFirewall, μ_2 is the processing rate of a cloud VM, λ_1 is the rate of the requests at the beginning of the vFirewall, and λ_2 is the rate of the total requests arriving at the cloud resources.

Sqalli et al. [5] have computed the resource utilization (U) at the cloud resources as follows:

$$U = \frac{\lambda_2}{S.\mu} \quad (4.6)$$

To measure the accuracy of EDoS-Shield validation we compare the results from Sqalli et al. [5] code with the simulation results that we got from the CloudSim simulator. The relative error percentage can be expressed as follows:

$$Relative\ error = \left|\frac{Results_{\text{cloudSim}} - Results_{\text{Sqalli et al. [5]}}}{Results_{\text{cloudSim}}}\right| * 100\% \quad (4.7)$$

The elasticity feature is one of the important characteristics of the cloud that allows the scaling up or down of the cloud resources based on some metrics specified by the cloud provider. One of these metrics is related to the optimization of the scaling VMs size value. Al-Haidari et al. [20] presented an optimization problem regarding the scaling of VMs size value. They concluded that the optimal number of VMs that should be added in one

provisioning period is 2. Another important metric is related to the tuning of the utilization upper threshold. An optimization problem regarding the utilization upper threshold was done also by Al-Haidari et al. [20]. Al-Haidari et al. concluded that the optimal upper utilization for cloud environment is 80%. Moreover, Al-Haidari et al. have calculated a formula for the minimum number of operational VMs to ensure that the average utilization remains below the upper threshold. The number of the required VMs is formulated as follows [20]:

$$U = \frac{\lambda}{S.\mu} \leq 0.8 \rightarrow S = \left\lceil 1.25 * \frac{\lambda}{\mu} + 1 \right\rceil \quad (4.8)$$

4.3 EDoS-Shield Simulation Model

The proposed work presented by Sqalli et al. [5] is built using CloudSim simulator [25]. Table 4.1 shows a summary of the parameters that have been used while simulating this technique.

Table 4.1: EDoS-Shield mitigation technique simulation parameters [5].

Parameter	Value
Request Size [32]	580 bytes
Average VM/Vnode Capacity	100 Req./Sec.
Average vFirewall Capacity	9260 Req./Sec.
Load Balancing algorithm	Round Robin
Legitimate Request rate (Fixed)	400 Req./Sec.
Attacker request rate (variable)	400-8000 Req./Sec.

The average capacity of the vFirewall can be calculated as follows [5]:

Average service time of vFirewall = average processing time for a request in the device driver + average processing time in the rule set $= \frac{reqest\ size\ (bits)\ [32]}{Max\ capacity\ of\ the\ insance\ (Mbps)[35]} + average\ processing\ time\ per\ rule\ [34] * \max sise\ of\ the\ rule\ set[36] = \frac{580*8}{400*10^6} + 2671 * 0.036\mu s = 11.6\mu s + 0.962ms = 108\mu s$

Average capacity of one vFirewall $= \frac{1}{Average\ service\ time\ of\ vFirewall} = \frac{1}{108\mu s} \approx 9260\ Req/sec.$

In addition, the average cloud response time can be calculated based on equation (4.5) by considering the following settings:

1. The capacity of the link between vFirewall and VMs assumed to be 10Gbps.
2. The number of initial running instances is 6.
3. The provisioning overhead is 55.4sec [37].
4. The upper utilization threshold is 80%.
5. The scaling size parameter is considered to be 2VM/provisioning period.

4.4 Enhanced EDoS-Shield Analytical Model

In this section we present the analytical model that was used to build the Enhanced EDoS-Shield mitigation technique. Al-Haidari et al. [6] have used the same queuing model of the EDoS-Shield [5] to model the Enhanced EDoS-Shield with minor modifications to the modeling of the vFirewall and vNode components.

Al-Haidari et al. [6] have evaluated their proposed technique in two scenarios: the whitelist scenario and the blacklist scenario. In the whitelist scenario, the attacker initially sends one legitimate request toward the cloud in order to add the attacker IP address to the whitelist. Then, the attacker controls a number of zombie machines while changing all their IP addresses to the attacker's whitelisted IP address. On the other hand, the blacklist scenario occurs when one of the legitimate clients sends requests towards the cloud but, unfortunately, the IP address of this legitimate user was used by an attacker and is already placed in the blacklist.

In the EDoS-Shield, the false positive rate that is associated with the attack rate that may pass the vFirewall is zero as Al-Haidari assumes that the EDoS-Shield is protected against the IP spoofing attacks. On the other hand, the Enhanced EDoS-Shield false positive rate will dramatically affect the results of the analytical method.

For the whitelist scenario, the percentage of the false negative is computed as follows:

$$P_{\text{False Negative}} = m * \frac{1}{255} * \frac{Z}{255} \tag{4.9}$$

Where m is the number of zombie masters, Z is the zombie machine.

For the blacklist scenario, the percentage of the false positives is computed as follows:

$$P_{\text{False Positive}} = m * \frac{1}{255} \tag{4.10}$$

In Table 4.2 we summarize all the equations used while calculating the performance metrics for the Enhanced EDoS-Shield mitigation technique.

Table 4.2: A summary of the Enhanced EDoS-Shield analytical model equations [6].

Metric	**Equation**
Response Time	$RT = \frac{S_1}{S_1.\mu_1 - \lambda_1} + \left(1 - \frac{\lambda_1}{\mu_{\text{link2}}}\right) / (\mu_{\text{link2}} - \lambda_1) + \frac{S_2}{S_2.\mu_2 - \lambda_2}$
Average Utilization	$\frac{\lambda_2}{S.\mu}$
Description: μ_{link2} is the capacity of the link from a vFirewall to the cloud VMs. S_1 is the number of the instances representing the vFirewall. μ_1 is the processing rate of a vFirewall. μ_2 is the processing rate of a cloud instance. λ_1 is the legitimate rate plus the false positive rate of the requests at the beginning of the vFirewall. λ_2 is the legitimate rate plus the false positive rate of the requests that arrived at the cloud VMs.	

4.5 Enhanced EDoS-Shield Simulation Model

The proposed work presented by Al-Haidari et al. [6] was built using the CloudSim simulator. All the simulation parameters for the EDoS-Shield discussed in section 4.3 were used in simulating the Enhanced EDoS-Shield. In addition, the following parameters are also used in the simulation:

1. The maximum value of the counter of unmatched TTL is set to be 5 as a previous study shows that about 95% of the network paths had fewer than 5 observable daily changes [39].
2. The attack lifetime is set to be 1 hour as around 90% of the cloud attacks did not exceed 1 hour [40].
3. The TTL values changes from 1 to 255.
4. Fixed Legitimate Request rate = 400 Req./Sec.
5. Variable attack request rate from 400-8000 Req./Sec.

4.6 Controlled Virtual Resources Access EDoS Mitigation (Analytical Model)

This section presents the analytical model that was used by Baig et al. [14] in order to build the Controlled Virtual Resources Access EDoS mitigation technique. Baig et al. proposed a rate limit algorithm to be used in the VM Investigator in order to mitigate the EDoS attack. The rate limit algorithm is based on the "number of accesses to give" (w_{opt}) parameter that is generated by the VM Investigator. The purpose of the w_{opt} parameter is to avoid flooding the VM Investigator resources by the user requests either through flash overcrowd or EDoS attacks. The w_{opt} parameter represents an upper bound on requests that the user can send in a limited time.

When the VM investigator receives a request from either the vFirewall or the VM Observer, the VM investigator sends a Turing test toward the owner of that request. If the owner of the request (i) fails to respond to the test, the VM investigator stores the following state information of that user:

State_information = {start time t_{i}, end time $t_{\text{i+1}}$, Max access counter w_{opt}, source IP address, time request received t_{s}};

Where:

- $1 \leq w_{\text{opt}} \leq \frac{L_{\text{i}}.v}{n}$, where L_{i} is the length of the buffer at the VM investigator that should hold all the requests from cloud user (i), v is the number of the operational VMs at the cloud provider, and n is the maximum parallel end users that attempt to access the cloud.

- $t_{\mathrm{i}} = t_{\mathrm{w}} + \Delta$, where t_{i} is the beginning of the time frame for the access control, t_{w} is the waiting time of the request in the buffer to gain access, and Δ is the network delay between the end user and the cloud provider.
- $t_{\mathrm{i+1}} = t_{\mathrm{i}} + w_{\mathrm{opt}}.(\tau + 2.\Delta + \delta_{\mathrm{i}})$, where $t_{\mathrm{i+1}}$ is the ending of the time frame for the access control, τ is the sum of all buffers which exists in all the operational VMs in the cloud, δ_{i} is the estimate of the minimum interarrival delay between two successive requests from the same user (i).

The time frame of the access control ($t_{\mathrm{i+1}} - t_{\mathrm{i}}$) is depends on the access parameter (w_{opt}) given to the user. So for small number of accesses granted to the user the time frame will be small, and vice versa. If a large number of access is granted to a given user, then this will be unfair for other end users, and the quality of experience for these users will be affected. On the contrary, if a small number of access is granted to a given user or users who access the cloud resources frequently, then this will cause a service disruption at the VM investigator, and a possible DoS may occur at the VM investigator. In order to solve this problem, Baig et al. [14] optimize the number of accesses which will be granted to the end user as follows: The authors calculate the total cost incurred at the service provider when the rate access scheme is operational, as follows:

$$C_{\mathrm{total}} = \frac{C_1 \,.\, A}{w_{\mathrm{opt}}} + c_2 \,.\, (1 - i).\, w_{\mathrm{opt}} \qquad (4.11)$$

Where

- C_1 is the cost of communication between the VM investigator and the end user (i.e. the cost of the mitigation technique).
- A is the estimate access counter for a given cloud resource C_{s}.

- C_2 is the cost of the under-utilized allocated time that is being granted to the malicious user by the cloud provider, which will affect the resource utilization at the cloud service provider (i.e. the cost of the EDoS attack).
- i is the fraction of the legitimate users that access the cloud where 0< i <1.

By minimizing the total cost in equation 4.11, we can figure out an optimal value for w_{opt} which will be as follows:

$$w_{\text{opt}} = \sqrt{\frac{C_1.A}{C_2.(1-i)}} \tag{4.12}$$

Based on the formula (4.12), the value of $t_{\text{i+1}}$ can be calculated, and subsequently, the value of t_{i}.

4.7 Controlled Virtual Resources Access EDoS Mitigation (Simulation Model)

The proposed work presented by Baig et al. [14] is built using the CloudSim simulator to evaluate the effect of the EDoS attack against the cloud provider services when the rate limit algorithm is being deployed. Table 4.3 shows the parameters that have been used while simulating this technique.

Table 4.3: Baig et al. [14] simulation parameters.

Parameter	Value
L_{i}	100
δ_{i}	10ms
A	100
C_1	0.5
C_2	0.5
Default UTF	0.5

4.8 Controlled Access to Cloud Resources EDoS Mitigation (Simulation Model)

In this section we present the simulation model that was used to build the Controlled Access to Cloud Resources EDoS Mitigation technique [21] in the CloudSim simulator while the "limited access permission" algorithm is being deployed. Table 4.4 shows the parameters that have been used while simulating this technique.

Table 4.4: Baig et al. [21] simulation parameters.

Parameter	Value
Legitimate load	180 Request/Sec. [41]
Web server type	Apache
UTF	Good (0.75,1] Average [0.25,0.75] Bad [0,0.25)
VM instance OS	CenOS 5.6 (64-bit)
Max VM instances	10
Min VM instances	3
Average VM Capacity	150 Request/Sec. [21]
Provisioning overhead	60s [33]
The upper threshold utilization	60s [33]
Auto Scaling upper Threshold	80% [33]
Auto Scaling lower Threshold	30% [21]
Auto Scaling metric	Utilization

CHAPTER 5 SIMUALTION VALIDATION RESULTS

In this chapter we present a thorough simulation validation for the approaches presented in [5] [6] [14] [21]. From the literature review, we found that these four approaches represent the most detailed mitigation techniques for protecting cloud services against the EDoS attack. Specifically, these approaches provide proper description of the system architecture, and present the associated performance results. While validating the aforementioned solutions, we consider the following metrics: the utilization of the computing resources, and the cloud response time. We perform the simulation validation under one simulation platform that was built using the CloudSim simulator so as to come up with consistent results for such techniques.

5.1 EDoS-Shield Validation

While validating the EDoS-Shield all the parameters and the details of Sqalli et al. work [5] were carefully followed. So, for the validation purpose we compare the results of the EDoS-Shield CloudSim simulator that we built with the results of Sqalli obtained code [5]. More specifically, we plot the response time and the cloud resources utilization for the cases when an EDoS attack is occurring while the mitigation technique is inactive as well as being active.

5.1.1 Response Time

Figure 5.1 shows that both simulations have almost similar results for the response time, the small variation is due to the randomness of both simulations. While the EDoS-Shield being inactive, both results show that when the attack traffic increases the cloud response time also increases. Also, it is clear that when the attack traffic is significantly increases, the response time does not increase by the same trend. This is due to the auto scaling algorithm that allocates more VMs to process the high load caused by the attack traffic. In general, the attack traffic results in an increase in the response time of the legitimate users when compared with the response time of having only 400 request/Sec. from legitimate users. On the other hand while the EDoS-Shield being active, the corresponding response time is approximately constant when the attack traffic increases. This is due to successfully blocking the attack requests from reaching the protected cloud service.

Figure 5.2 shows the relative error between the CloudSim EDoS-Shield results and the corresponding result of Sqalli et al. work [5]. The relative error is computed using equation (4.7). Figure 5.2 shows that the relative error percentage does not exceed 0.6573%, which indicates that the CloudSim Simulation has a good accuracy.

5.1.2 Utilization

Figure 5.3 presents the cloud resources utilization for both the CloudSim EDoS-Shield simulation and Sqalli work [5]. The Figure shows that the two results are identical with a maximum relative error 0.141% as observed from Figure 5.4. In Figure 5.3 while the EDoS-Shield being inactive, the average cloud resources utilization has a similar trend to the results obtained for the cloud response time, where the utilization increases whenever there is an increase in the attack rate. So in general, the EDoS attack consumes more cloud computational power than when there is no attack. On the other hand while the EDoS-Shield being active, the average cloud resources utilization is not affected due to the attack rate as the attack requests will not reach the target cloud service.

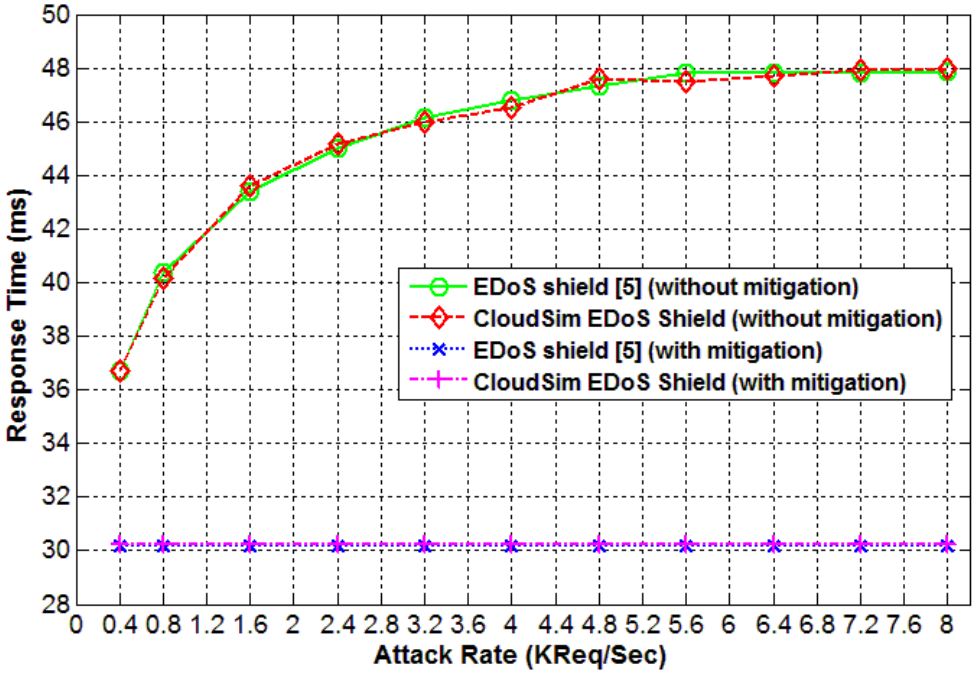


Figure 5.1: Response time results for EDoS-Shield [5] and EDoS-Shield CloudSim.

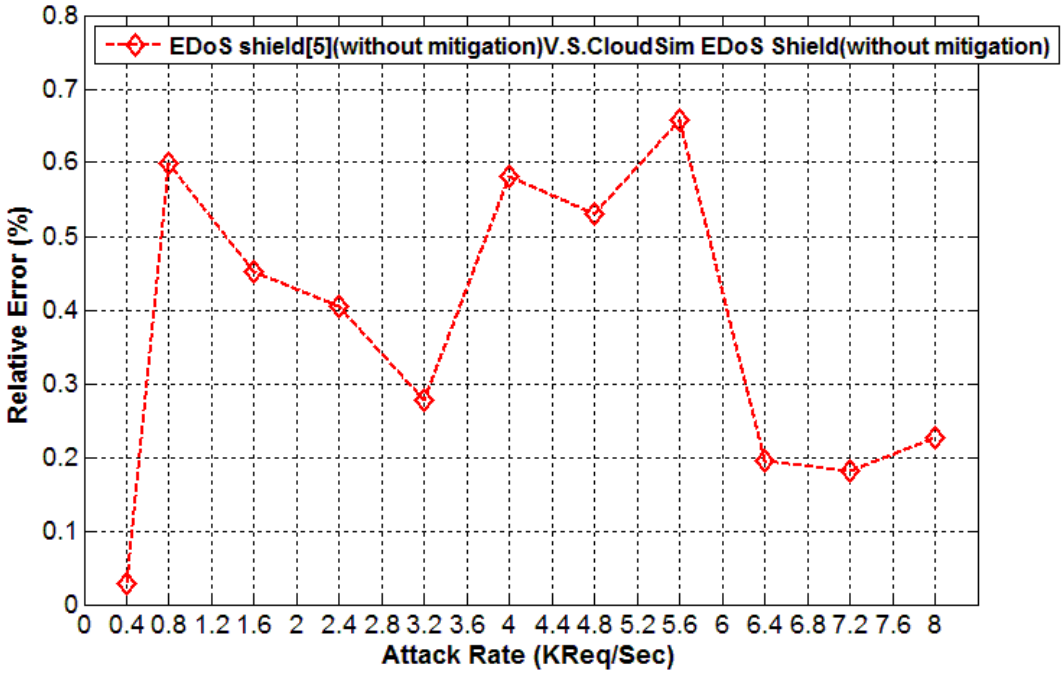


Figure 5.2: Response time relative error percentage for EDoS-Shield [5] and EDoS-Shield CloudSim.

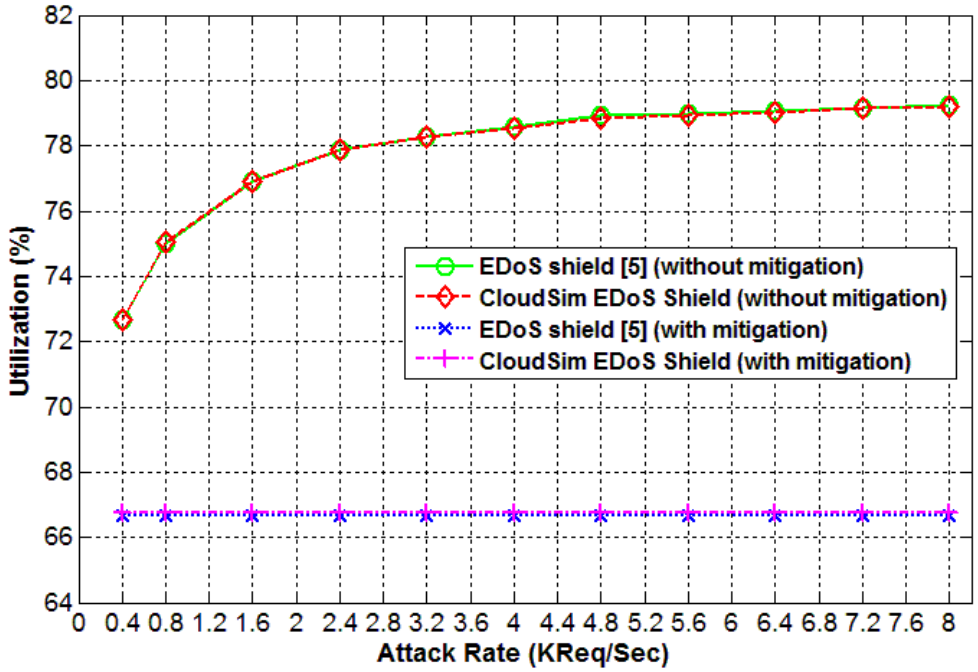


Figure 5.3: The computing resources utilization results for EDoS-Shield [5] and EDoS-Shield CloudSim.

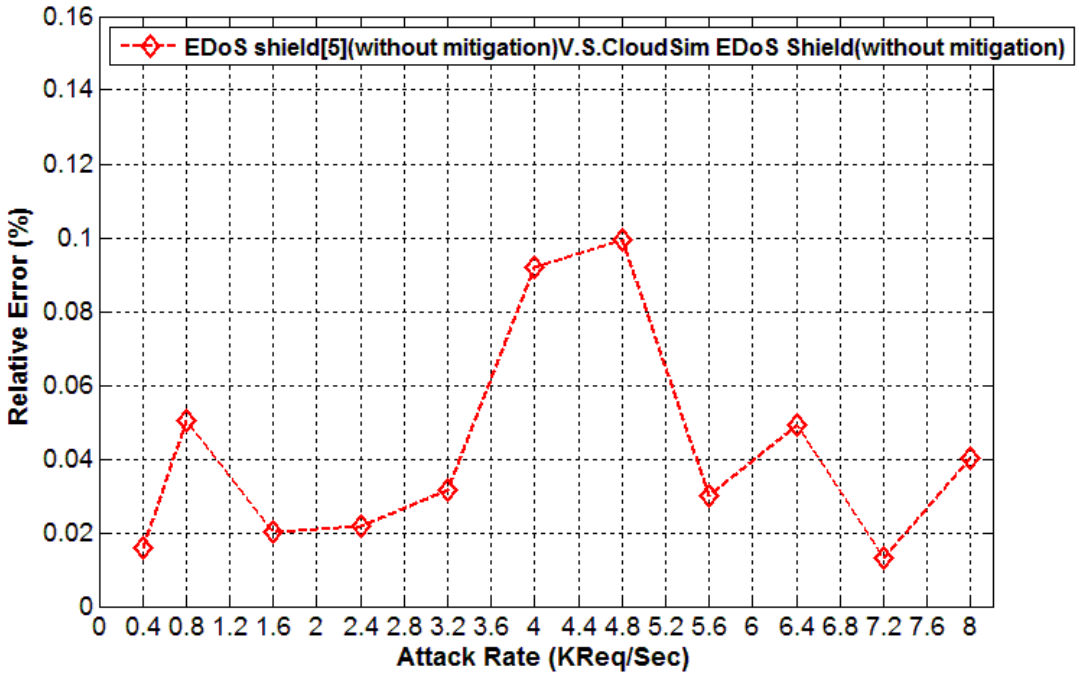


Figure 5.4: The computing resources utilization relative error percentage for EDoS-Shield [5] and EDoS-Shield CloudSim.

5.2 Enhanced EDoS-Shield Validation

While validating the Enhanced EDoS-Shield we used all the parameters and the details of Al-Haidari work [6]. So, for the validation purpose we compare the results of the Enhanced EDoS-Shield CloudSim simulator with the results from the obtained Al-Haidari code. For the purpose of validation, the response time and the cloud resources utilization for both the blacklist case and the whitelist case are plotted and compared with the results provided in [6].

5.2.1 Blacklist case results

In the Enhanced EDoS-Shield blacklist case, initially the Enhanced EDoS-Shield detects number of attackers that carry out an attack using spoofed IP addresses. Accordingly, the Enhanced EDoS-Shield places the IP addresses and TTL values of these attackers in the blacklist. A problem might occur when a legitimate user tries to send a request towards the cloud using an IP address that is already blacklisted. Consequently, the Enhanced EDoS-Shield will drop that request and block the legitimate user from being served.

From Figures [5.5-5.6] it is observed that the obtained response time simulation results for Enhanced EDoS-Shield [6] and CloudSim Enhanced EDoS-Shield are close to each other with a relative error percentage of 0.4202%. Also, From Figures [5.7-5.8] it is clear that the obtained utilization simulation results for Enhanced EDoS-Shield [6] and CloudSim Enhanced EDoS-Shield are close to each other with a relative error percentage of 0.1669%. The results in Figure 5.5 and Figure 5.7 show a decrease in the cloud response time and the average cloud resources utilization as compared with the no attack case. This is due to an assumption made by the authors that 20% of the total attacker's requests have the same IP address and TTL value as some legitimate users. In this case, all the legitimate users that

have the same IP address and TTL value as for the attacker will be dropped mistakenly by the Enhanced EDoS-Shield as they will be considered as attackers. Accordingly, less number of legitimate requests can access the cloud resources which results in decreasing the cloud response time and utilization.

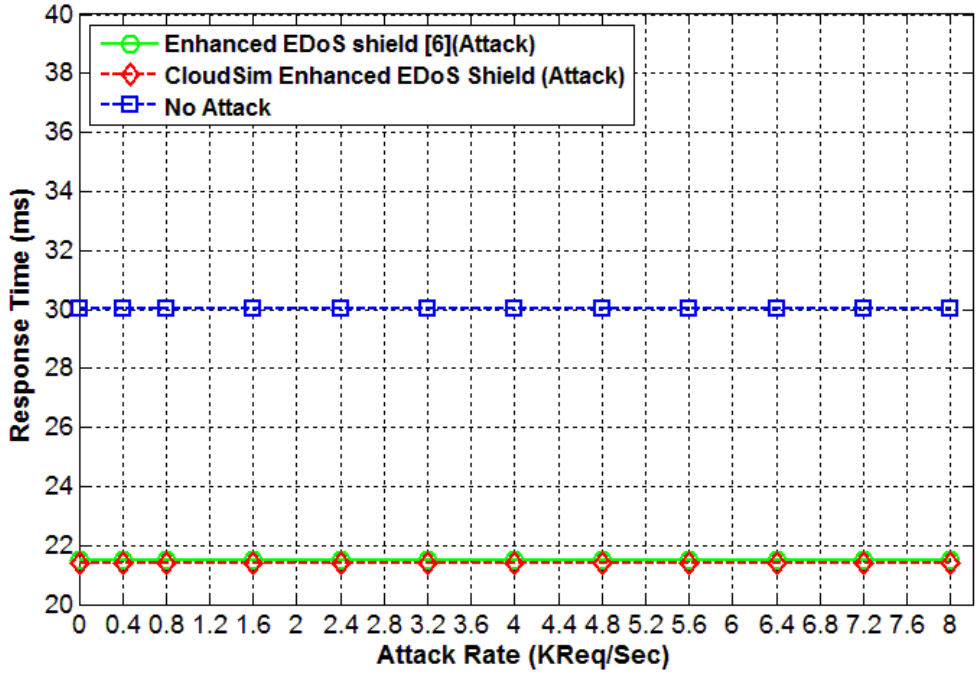


Figure 5.5: Response time results for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the blacklist case.

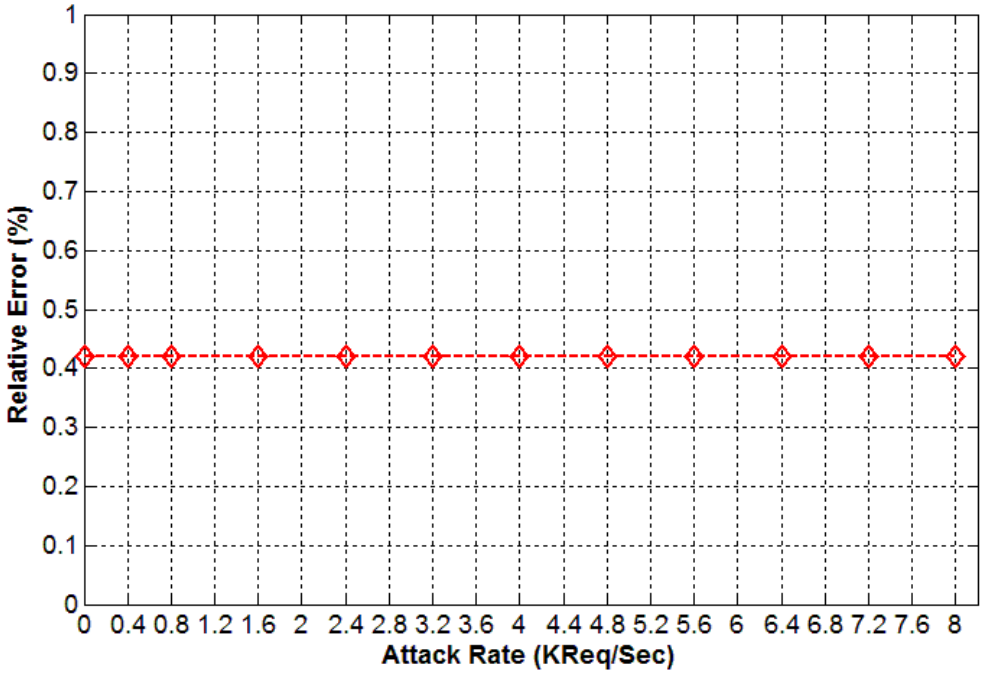


Figure 5.6: Response time relative error percentage for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the blacklist case.

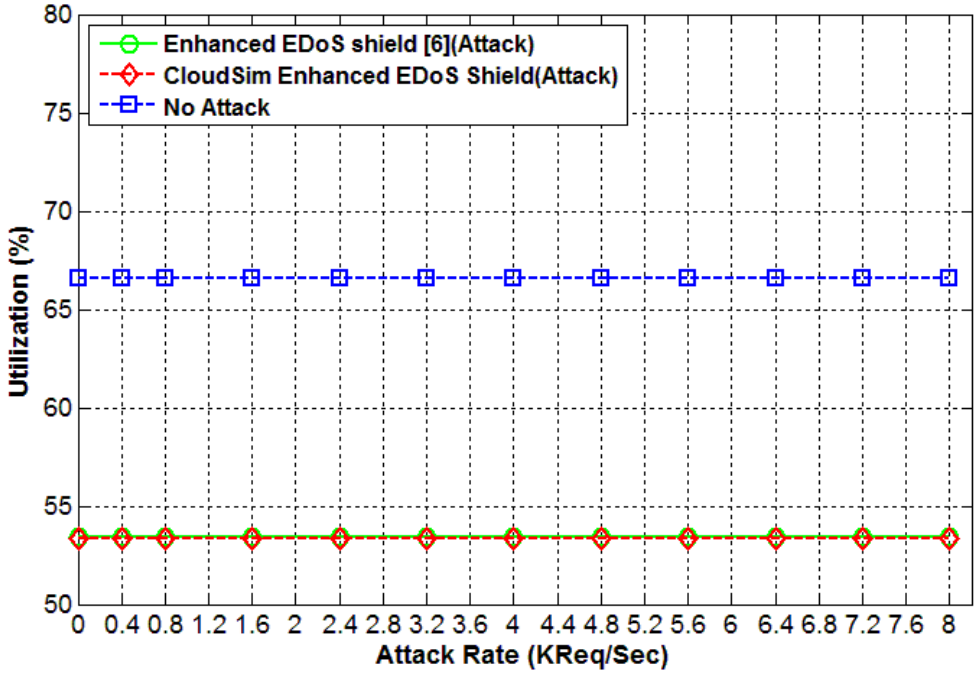


Figure 5.7: The computing resources utilization results for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the blacklist case.

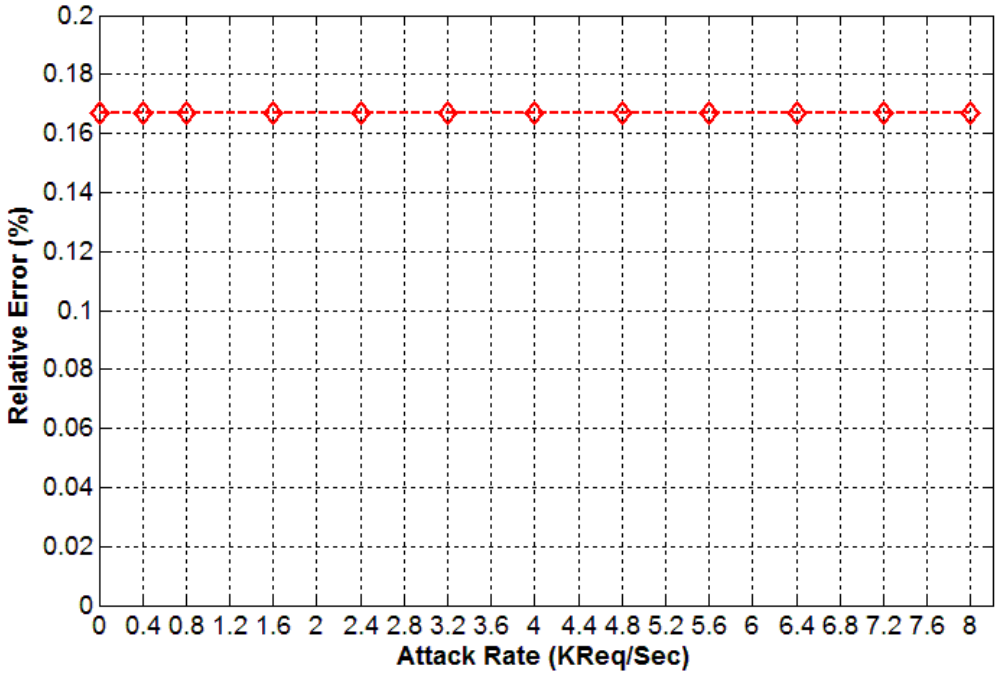


Figure 5.8: The computing resources utilization relative error percentage for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the blacklist case.

5.2.2 Whitelist case results

In the Enhanced EDoS-Shield whitelist case, the IP address of an attacker is placed in the whitelist by sending one legitimate request from the attacker's machine. Then, the attacker orders a set of bot machines to generate a huge amount of attack traffic toward the cloud while forging their IP addresses using the attacker whitelisted IP address. Subsequently, the corresponding traffic of these bot machines is forwarded to the cloud.

From Figures [5.9-5.10] it is clear that the obtained response time simulation results for Enhanced EDoS-Shield [6] and CloudSim Enhanced EDoS-Shield are close to each other with a relative error percentage less than 0.6868%. Also, From Figures [5.11-5.12] the obtained utilization simulation results for Enhanced EDoS-Shield [6] and CloudSim Enhanced EDoS-Shield are close to each other with a relative error percentage less than 0.3068%. Furthermore, it is observed from Figure 5.9 and 5.11 that when the attack rate is below the 4800 Req./Sec. the Enhanced EDoS-Shield shows a response time and utilization results less than the results obtained for the no attack case. This is due to an assumption made by the authors that 20% of the total bot's requests have the same TTL value as that of the attacker whitelisted TTL value (false negative percentage). In this case, all the bot requests that have the same IP address and TTL value as that of the attacker will be forwarded to the cloud by the Enhanced EDoS-Shield as such bot requests are considered to be legitimate requests. Therefore, the auto scaling algorithm in the cloud will be triggered in order to handle the bot's traffic that reaches the cloud. Since the false negative rate is proportional to the attack rate, the false negative excess traffic when the attack rate is below 4800 Req./Sec. will consume little computational power from the added VMs. This leads to having less average response time when compared with the no attack case.

Likewise, when the attack rate is above the 4800 Req./Sec., the false negative excess traffic saturates the added VM instances with requests and leads to having an average response time greater than the no attack case.

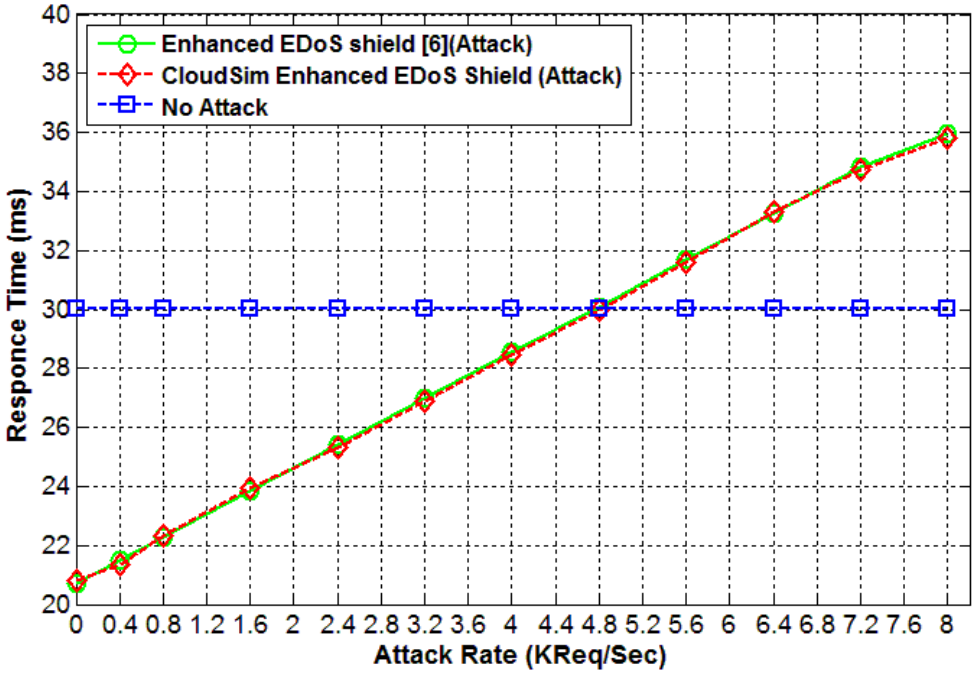


Figure 5.9: Response time results for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the whitelist case.

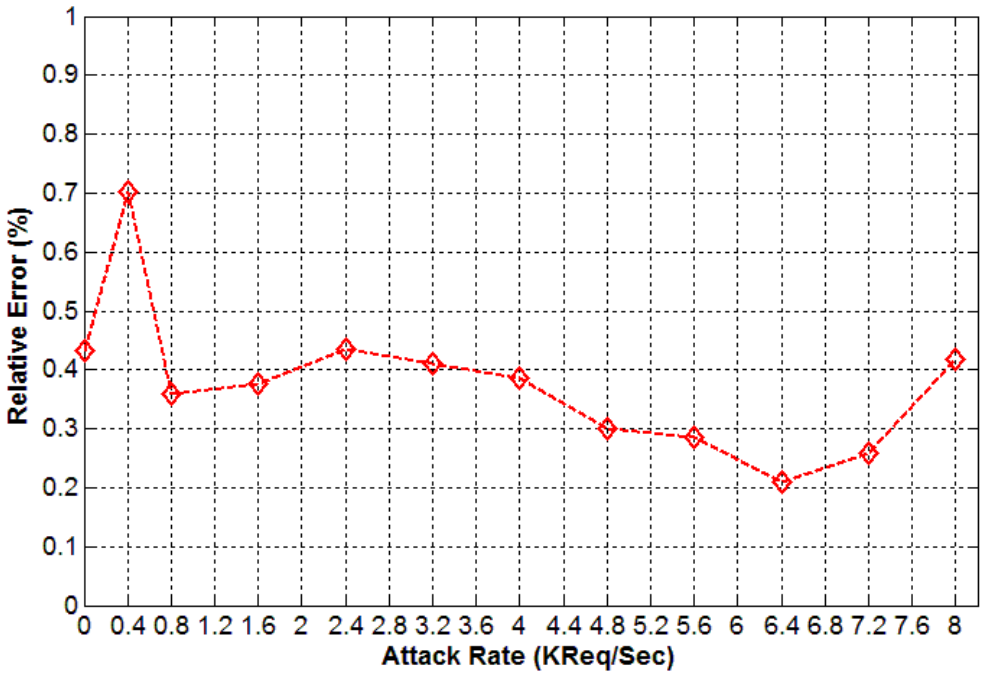


Figure 5.10: Response time relative error percentage for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the whitelist case.

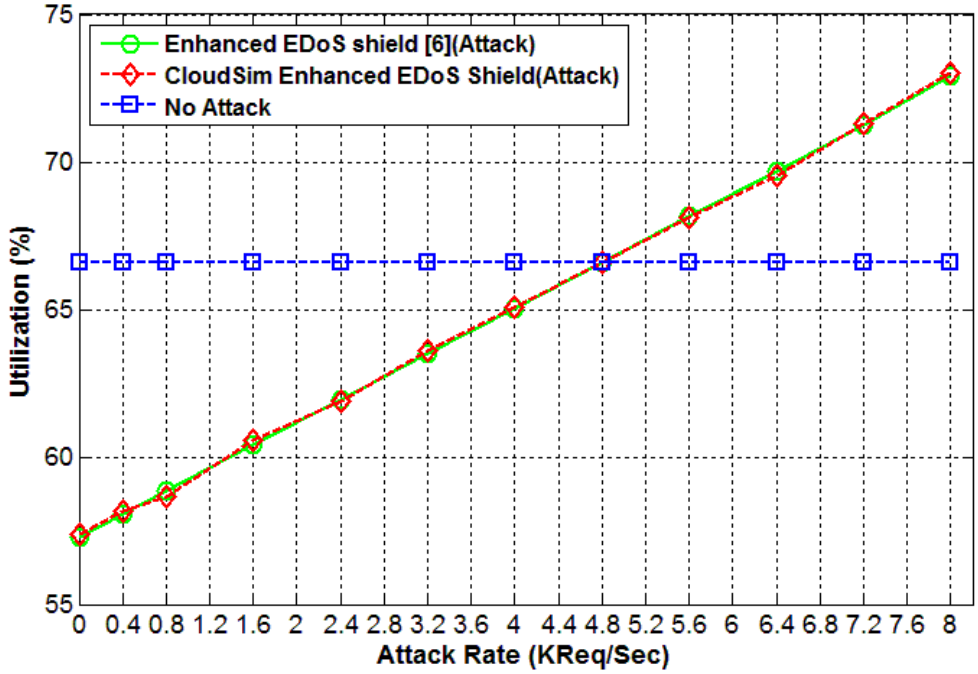


Figure 5.11: The computing resources utilization results for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the whitelist case.

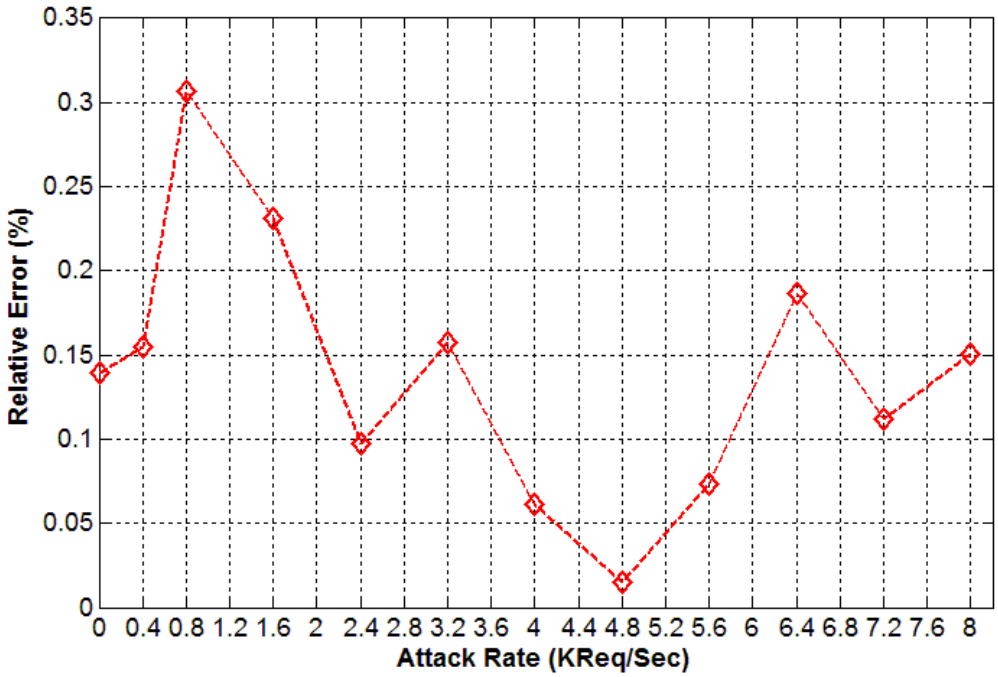


Figure 5.12: The computing resources utilization relative error percentage for Enhanced EDoS-Shield [6] and Enhanced EDoS-Shield CloudSim for the whitelist case.

5.3 Controlled Virtual Resources Access EDoS Mitigation Validation

In order to validate the Controlled Virtual Resources Access all the parameters and the details presented by Baig et al. [14] were followed. So, for the validation purpose we compare the results of the CloudSim simulator with the results from the obtained Baig et al. code [14]. For the purpose of validating the simulator, the response time metric is compared with the corresponding response time evaluated in [14] when the number of VMs is equal to 500.

5.3.1 Response Time

As evident from Figure 5.13 and Figure 5.14, there is a small difference between the results presented by Baig et al. [14] and the CloudSim Simulation results with a relative error percentage of at most 0.7791%. In [14], the analytical model was included in the code that generated the performance results. On the other hand, the CloudSim simulator mimics the whole cloud infrastructure. Subsequently, a small variation appears between the Baig et al. results and CloudSim simulation results.

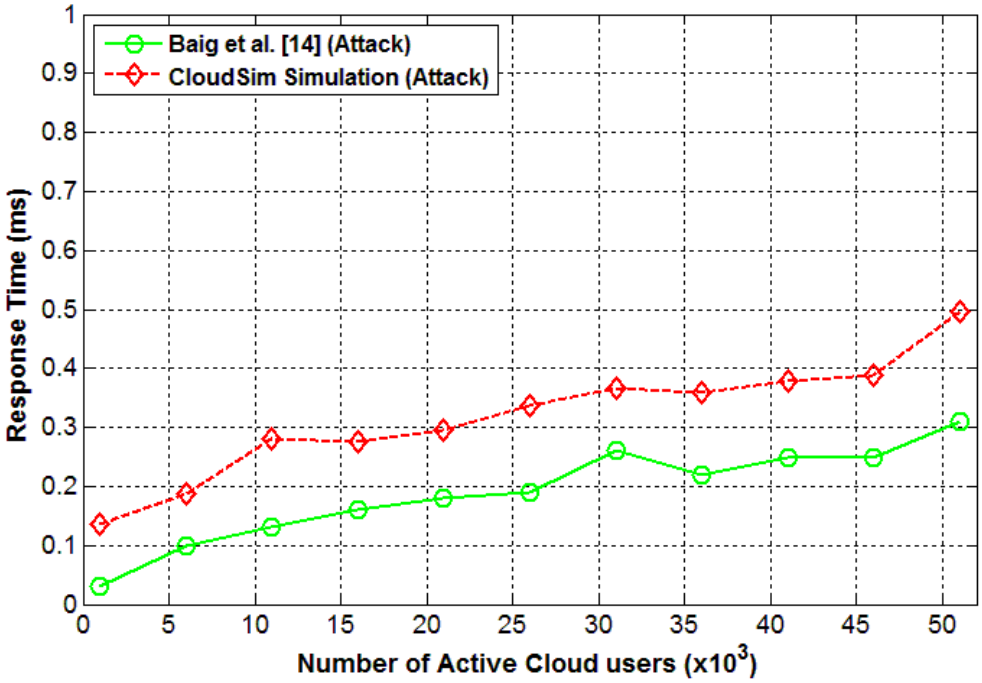


Figure 5.13: Response time results for Baig [14] and CloudSim simulation.

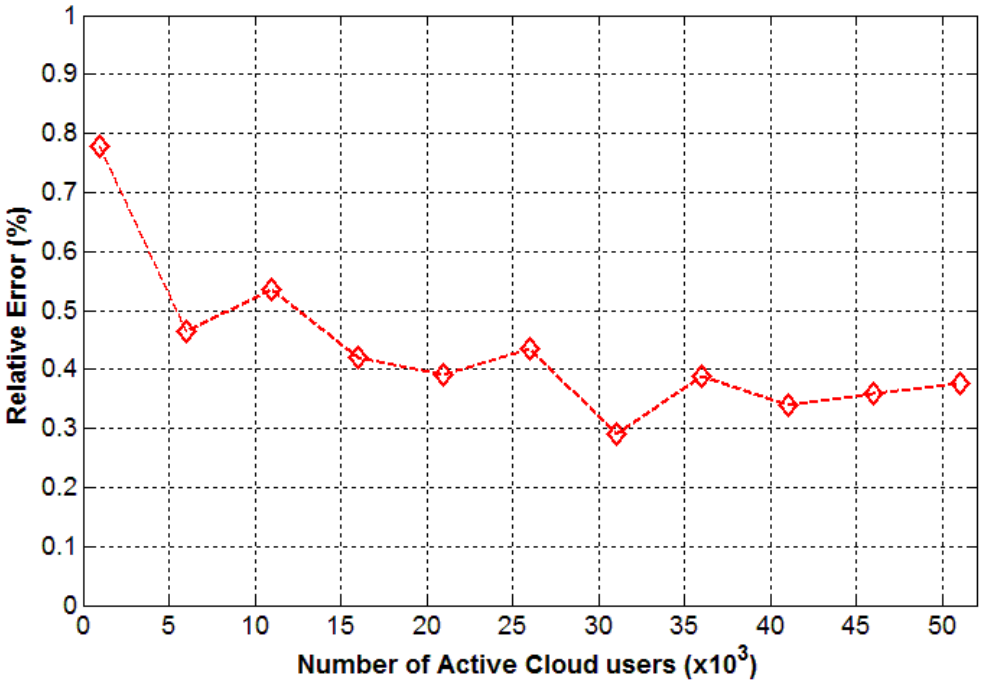


Figure 5.14: Response time relative error percentage for Baig [14] and CloudSim simulation.

5.4 Controlled Access to Cloud Resources EDoS Mitigation Validation

In order to validate the Controlled access to cloud resources EDoS mitigation, all the parameters and the details presented in [21] where followed. For the validation purpose, the response time and the cloud resources CPU utilization metrics obtained from the CloudSim simulator are compared with the corresponding metrics presented by Baig et al. [21].

5.4.1 Response Time

As shown in Figure 5.15 and Figure 5.16, there is a small difference between the results presented by Baig et al. [21] and the CloudSim Simulation results with a relative error percentage of less than 1.4761%. In [21], Baig et al. tested their mitigation technique using an experimental testbed and collected the performance results accordingly. Conversely, we collect the performance results using the CloudSim simulator. Subsequently, there is a small relative error between the CloudSim simulation results and the results presented by Baig et al. [21].

It is noted that, the response time while the attack rate being below 1000 request/Sec. roughly stays around 20ms as evident from Figure 5.15. This is mostly due to the auto scaling that allocates more VMs to accommodate all of the users demand. On the other hand, when the attacking traffic exceeds 1000 requests/Sec. the response time increases rapidly because the maximum VM instances of 10 VMs cannot instantaneously serve the excess traffic.

5.4.2 Utilization

Figure 5.17 and Figure 5.18 shows that the utilization results are close for the CloudSim simulation and Baig et al. testbed with a maximum relative error percentage of 1.2768%. Focusing on Figure 5.17, it is observed that the average CPU utilization of the cloud resources is 40% when the attack rate is zero. The 40% utilization is due to the fixed legitimate load. When the attack rate is 1200 Req./Sec. the analytical CPU utilization is equal to (attack rate + legitimate rate)/ capacity of the running VMs which is equal to (1200+180)/(10*150)=92%. Above this rate the system will be over saturated with requests and the average response time will increase dramatically as shown in Figure 5.15.

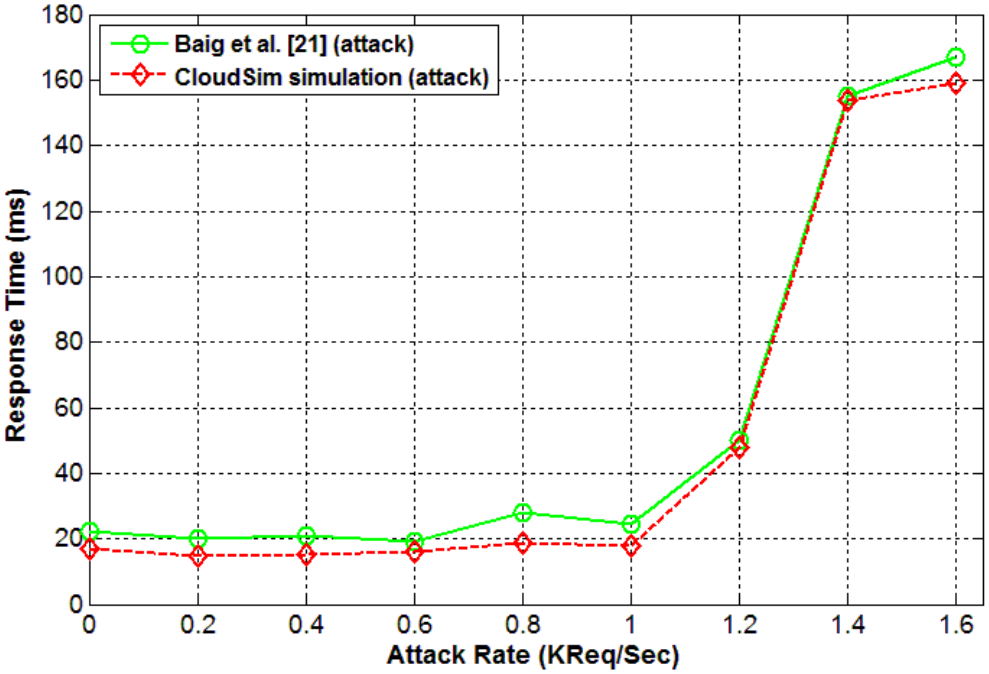


Figure 5.15: Response time results for Baig et al. [21] and CloudSim simulation.

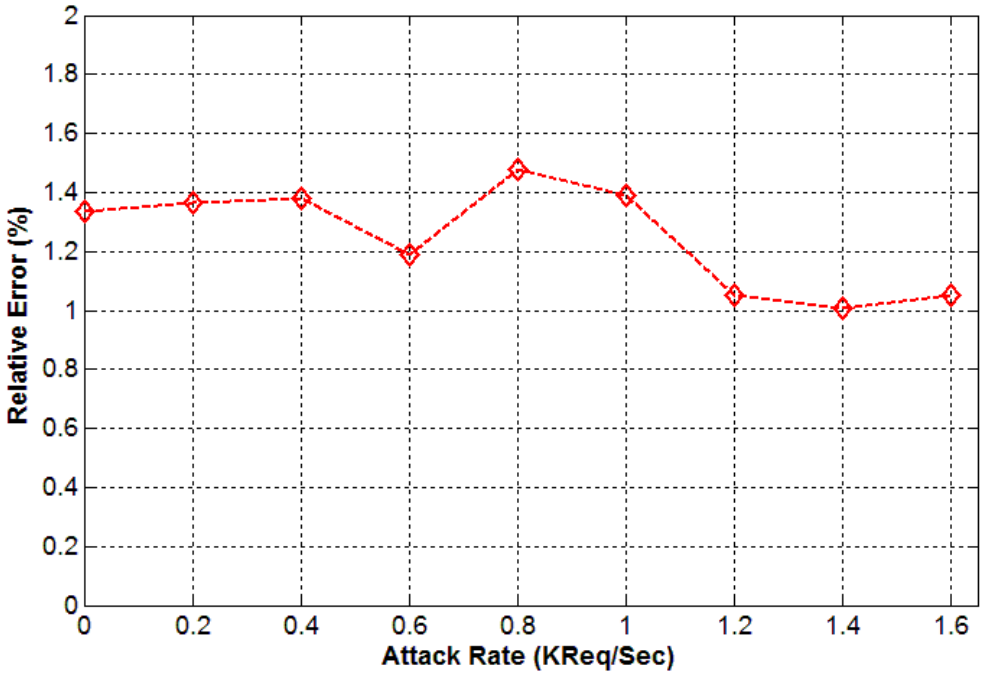


Figure 5.16: Response time relative error percentage for Baig et al. [21] and CloudSim simulation.

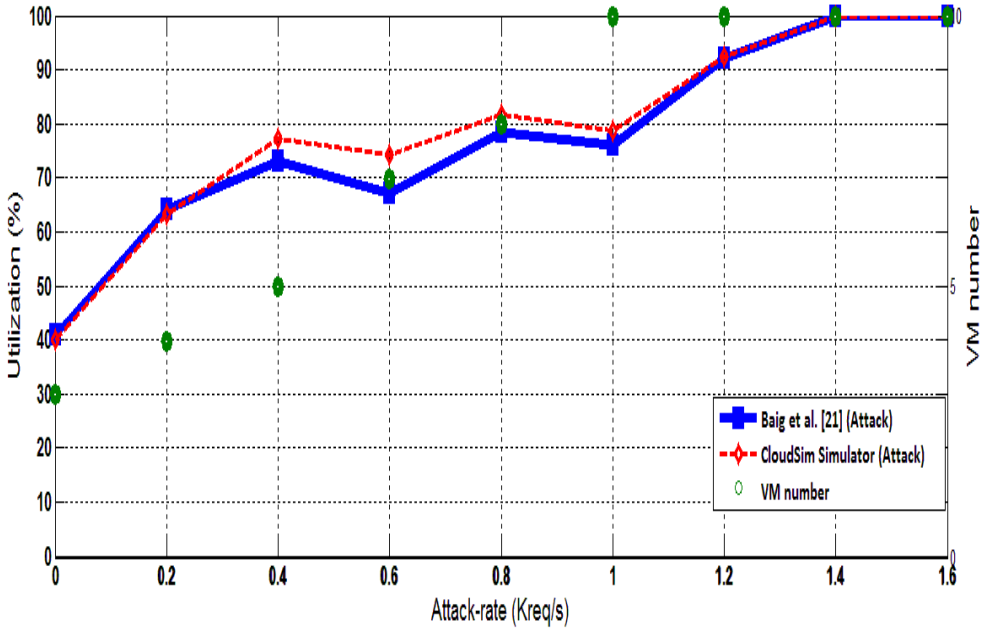


Figure 5.17: The computing resources utilization results for Baig et al. [21] and CloudSim simulation.

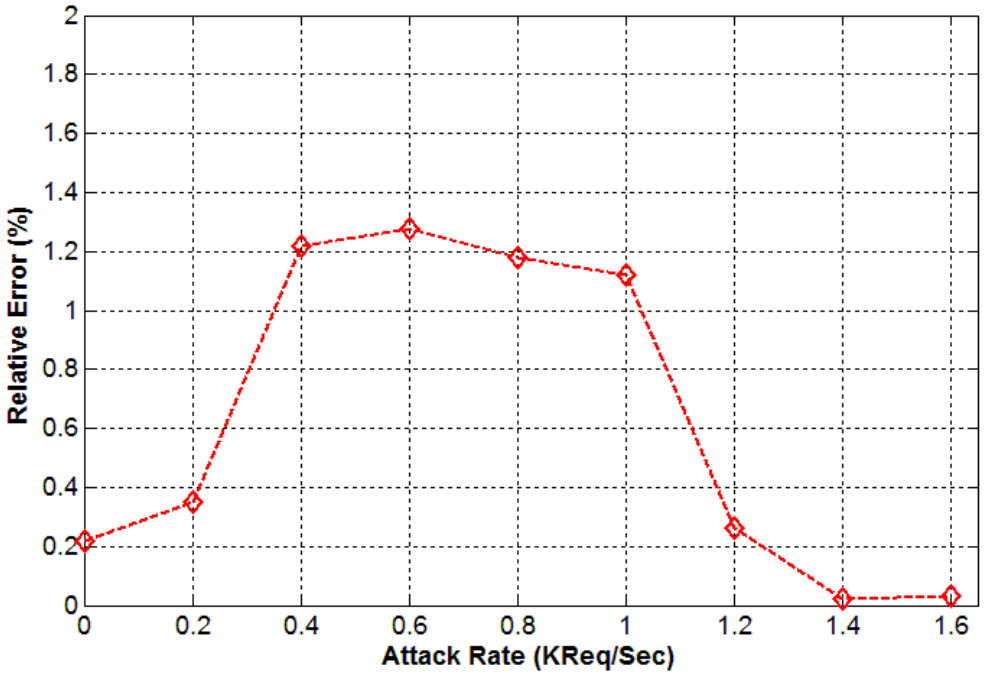


Figure 5.18: The computing resources utilization relative error percentage for Baig et al. [21] and CloudSim simulation

CHAPTER 6 PERFORMANCE SIMULATION RESULTS AND ANALYSIS

In this chapter, we present the performance simulation results of the EDoS mitigation techniques under study while considering the effect of different real cloud implementation conditions. We have conducted four experimental modes using the simulation models of the four mitigation techniques under study that were discussed in chapter 3 and chapter 4. The first mode is called the ***normal mode*** in which we expose the four mitigation techniques to different arrival rates of only legitimate user's traffic. The purpose of this mode is to examine if any of the EDoS mitigation techniques causes an overhead to the cloud service. The second mode is called the ***attack mode with IP spoofing users*** in which we consider two cases in order to cover all the possibilities of IP spoofing users. The two cases are the whitelist case and the blacklist case. This mode is highly needed to see if the EDoS mitigation techniques can handle smart attackers capable of tampering with the IP addresses before sending their requests to the cloud service. The third mode is called the ***attack mode with same NAT-based network users*** in which the cloud legitimate users and attackers belong to the same NAT-based network. Such mode reflects real life more than the previous modes since a considerable amount of the networks have their users behind a NAT router. The last mode is called the ***flash overcrowd mode*** in which an enormous amount of legitimate traffic is coming toward the cloud in a short period of time. This mode is also reflected in real life when a massive amount of traffic is generated in a short period due to some event such as a sports event or a sales promotion event. The successful EDoS

mitigation technique should be able to differentiate this behavior from an attack behavior and, accordingly, provides the same quality of service as in the normal mode.

In each of the aforementioned modes, we consider the effect of the following three real cloud implementation conditions:

1- The effect of using different load balancing algorithms: Beside the Round Robin (RR) algorithm that was used by all authors of the mitigation techniques under study, we consider the Least Loaded (LL) dispatching algorithm [42]. The LL makes the load balancer capable of sending the incoming request to the VM that has the lowest workload. In order to figure out who's VM has the lowest workload, a variable called "Finishing Time" is defined for each VM. This variable adds the service times of all the existing requests in a particular VM. Hence, the VM that has the lowest finishing time receives the next request from the load balancer.

2- The effect of using the Uniform Resource Locator (URL) redirection technique [43] to identify the automated attackers. Note that none of the EDoS mitigation techniques under study use this technique. The URL redirection technique can replace the CAPTCHA Turing test technique used by the EDoS mitigation techniques under study. The URL redirection technique transmits a URL redirection packet to the cloud user with a location field in the header that includes a virtual IP Address. This implies that the cloud user has to redirect the request to a different URL by using the received virtual IP address [43]. For a legitimate user the web browser can respond to the URL redirection packet without human intervention. On the other hand, an attacker usually runs a script on a bot machine to generate the attack traffic. Consequently, the attacker does not wait for the cloud response

packet. Therefore, the attacker does not react to the URL redirection. Accordingly, the benefit of using the URL redirection technique is reflected in reducing the average response time since the URL redirection overhead takes 0.63 seconds on average [44]. Consequently, the URL redirection overhead is approximately seven times less than the CAPTCHA Turing test overhead as reported in [45].

3- The effect of using different probability distributions for request service times for cloud users input traffic. Several studies have considered the exponential distribution to characterize the request service time in the cloud service [46-49]. On the other hand, others have shown that the heavy-tailed Pareto distribution can fit several essential characteristics of web servers and coincides with Internet traffic behavior [50, 51]. More importantly, Al-Fayoumi [52] uses Pareto distribution to model the service time of the cloud incoming traffic. Subsequently, the four mitigation techniques are evaluated using these two characterizations of input traffic.

Several researchers have modeled the cloud-based service as a network of queues and each VM is considered as a single queue [53, 54]. In the cloud service, there are usually multiple cloud servers employed to offer the service to cloud customers. Thus, parallel M/M/1 and parallel M/Pareto/1 queuing models are used in the evaluation process.

In order to simulate a full factorial experiment for the three aforementioned real cloud implementation conditions, five cases in the normal mode are considered. The first case uses the RR as a load balancing technique, the URL redirection as a Turing test, and the exponential probability distribution as an input traffic. The first case is referred to as simulation case 1: [RR-U-E]. The second case uses the RR as a load balancing technique,

the URL redirection as a Turing test, and the Pareto probability distribution as an input traffic. The second case is referred to as simulation case 2: [RR-U-P]. The third case uses the LL as a load balancing technique, the URL redirection as a Turing test, and the exponential probability distribution as an input traffic. The third case is referred to as simulation case 3: [LL-U-E]. The fourth case uses the LL as a load balancing technique, the URL redirection as a Turing test, and the Pareto probability distribution as an input traffic. The fourth case is referred to as simulation case 4: [LL-U-P]. Finally, the fifth case compares the best simulation case from these four cases with the author's mitigation technique simulation parameters of RR as a load balancing technique, CAPTCHA as a Turing test, and exponential probability distribution as an input traffic. The fifth case is referred to as simulation case 5: [RR-CAP-E]. The purpose of the fifth case is to find out if a variation of these parameters would provide for a better system performance than what is considered in the mitigation techniques original simulation results. In order to do so, we define the normalized response time and normalized CPU utilization as follows:

Normalized Response Time of the best simulation case = average response time of simulation case 5: [RR-CAP-E] / average response time of the best of the first four simulation cases. (6.1)

Normalized CPU utilization of the best simulation case = average CPU utilization of simulation case 5: [RR-CAP-E] / average CPU utilization of the best of the first four simulation cases. (6.2)

If the normalized value produces a one, it indicates that the two cases have the same performance.

In this chapter, the considered EDoS mitigation techniques are named as follows: the EDoS-Shield [5] is named (mitigation technique 1), the Enhanced EDoS-Shield [6] is named (mitigation technique 2), the Controlled Virtual Resources Access [14] is named (mitigation technique 3), and the Controlled Access to Cloud Resources [21] is named (mitigation technique 4).

Table 6.1 summarizes all the performance evaluation modes that are presented in this chapter. Four experimental modes are conducted using the simulation models discussed in chapter 3 and chapter 4.

Table 6.1: Performance Evaluation Modes.

Modes	Mitigation techniques	Simulation cases
1- Normal mode	(1)(2)(3)(4)	Case 1: RR,URL redirection, Exponential. Case 2: RR,URL redirection, Pareto. Case 3: LL,URL redirection, Exponential. Case 4: LL,URL redirection, Pareto. Comparison between best case & case 5: RR, CAPTCHA, Exponential.
2-Flash Overcrowd mode	(1)(2)(3)(4)	Comparison between case 3 & case 5
3- Attack mode (IP Spoofing) ❖Blacklist case ❖Whitelist case	(1)(2)(3)(4)	=
4- Attack mode (Behind the NAT) ❖Blacklist case ❖Whitelist case	(1)(2)(3)(4)	=

Table 6.2 summarizes the parameters used in the simulation. All the parameters are picked from the original work of the four mitigation techniques under study. Some parameters are common in the four mitigation techniques while others are specific to one or two mitigation technique(s).

Table 6.2: Simulation Parameters.

Parameter	Value	Reference
VM instance type	Small	[5,6,14,21]
Load Balancer instance type	Large	[5,6]
vFirewall instance type	Large	[5,6]
Scaling-up upper threshold	80%	[5,6,14,21]
Scaling-down lower threshold	30%	[5,6,14,21]
Auto scaling metric	CPU usage	[5,6,14,21]
Initial running servers	6	[5,6]
Scaling size parameter (Mitigation 1 & 2)	2	[5,6]
Scaling size parameter (Mitigation 3 & 4)	1	[14,21]
Provisioning Overhead (Mitigation 1 & 2)	55.4sec	[5,6]
Provisioning Overhead (Mitigation 3 & 4)	60sec	[14,21]
Upper threshold duration (Mitigation 1 & 2)	5 min	[5,6]
Upper threshold duration (Mitigation 3 & 4)	1 min	[14,21]
Counter of unmatched TTL (max value) (Mitigation 2)	5	[6]
Attack life time (Mitigation 2)	60 min	[6]
Default UTF (Mitigation 3 & 4)	0.5	[14,21]

6.1 Normal Mode Results

In this mode, the cloud service has enough VMs to handle the incoming legitimate traffic without the need for auto scaling. The arrival legitimate rate is varied from 400 Req./Sec. to 8000 Req./Sec. Also, the cloud does not receive any type of attack traffic. The objective of this mode is to see if the mitigation techniques under study can handle the normal load without adding overhead to the cloud service. In this section, the cloud response time and the resource utilization simulation results of the four mitigation techniques under study are presented. The results take into account changing the load dispatching algorithm, the algorithms that can identify the automated attackers, and the probability distributions for request service time for cloud users input traffic. Hence, the results are for simulation cases 1 through 5.

6.1.1 Response Time and Utilization Results for Simulation case 1 [RR-U-E]

The average response time results for the four mitigation techniques under study after applying the parameters of case 1 [RR-U-E] are depicted in Figure 6.1. The average cloud response time of the mitigation techniques (3) and (4) are the same as these two techniques are classified as reactive schemes. As such, these two techniques operate only when certain conditions are met. Specifically, if the cloud average CPU utilization exceeds 80% threshold or if the request comes from a blacklisted user then the technique is invoked. In the normal mode, the two techniques are not invoked as there are enough VMs to serve all legitimate requests without crossing the CPU utilization threshold. Also, none of the cloud legitimate users exist in the blacklist. Subsequently, the dispatcher will forward all the requests to the cloud VMs without interfering. Thus, these two techniques do not generate any overhead while serving legitimate requests. On the other hand, mitigation techniques (1) and (2) provide a noticeable overhead as compared to mitigation techniques (3) and (4). This is due to the fact that the vFirewall of mitigation techniques (1) and (2) checks the IP addresses of the incoming requests and sends Turing tests to the users to classify the users into whitelist and blacklist.

Figure 6.2 shows the average resources utilization results for the four mitigation techniques under study after applying case 1 [RR-U-E] parameters. All of the mitigation techniques give identical results for the average resource utilization as each server in the cloud serves equal amount of requests.

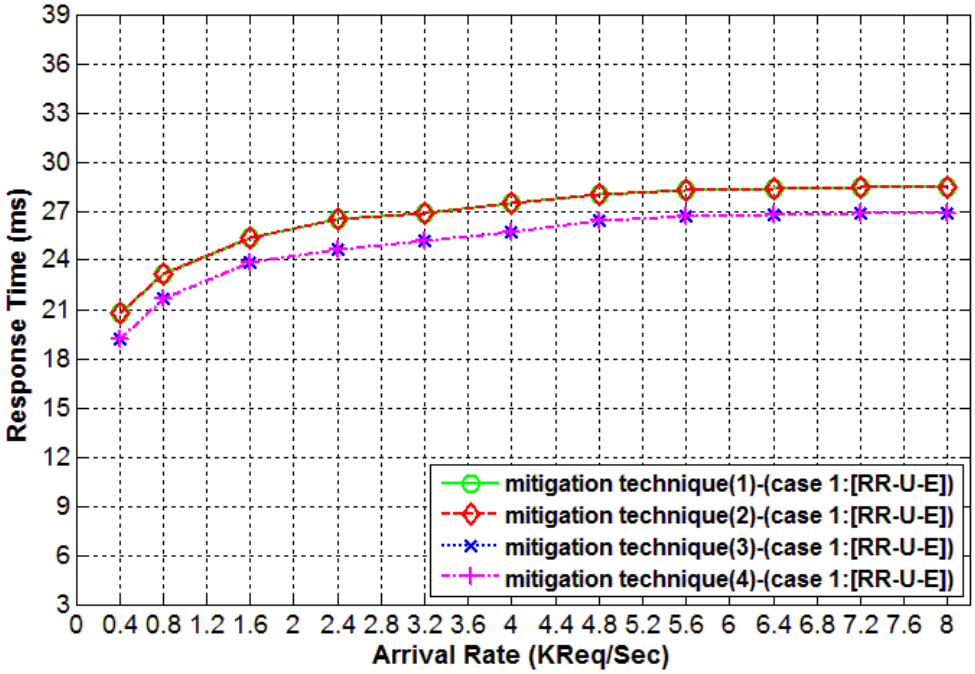


Figure 6.1: Response time result in the normal mode after applying case 1 [RR-U-E].

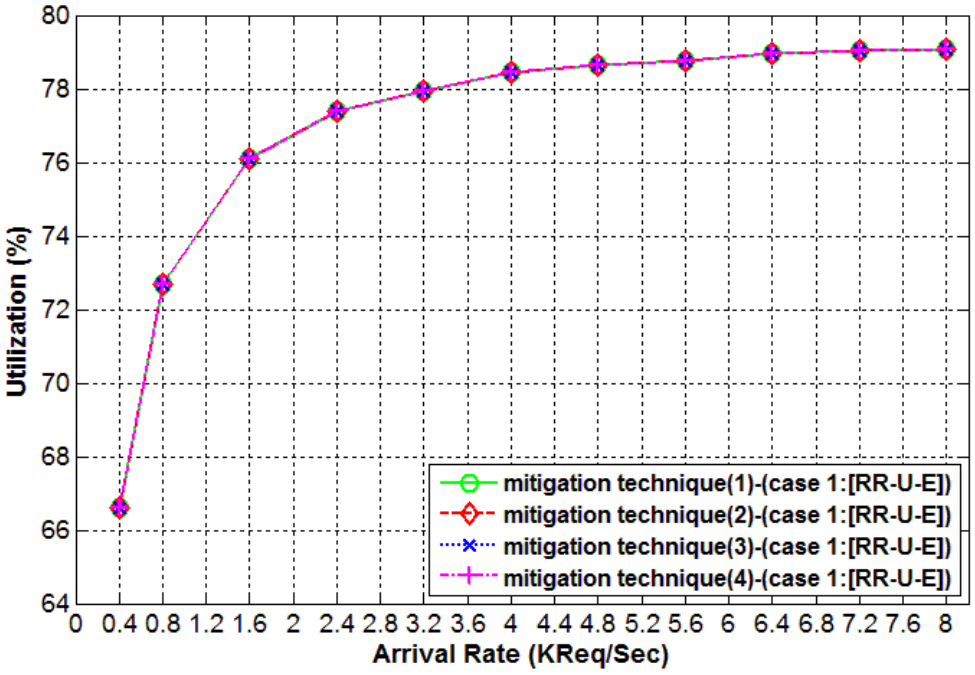


Figure 6.2: Resource Utilization result in the normal mode after applying case1 [RR-U-E].

6.1.2 Response Time and Utilization Results for Simulation case 2 [RR-U-P]

Figure 6.3 shows the average cloud response time results for the four mitigation techniques under study while considering the parameters of case 2 [RR-U-P]. The cloud response time results for mitigation techniques (3) and (4) are identical for the same reason as stated in section 6.1.1. Subsequently, these two techniques do not generate any overhead while serving legitimate requests.

Comparing Figures 6.1 and 6.3, there is a noticeable increase in the cloud response time results for case 2. This increase is due to the heavy right tail of the Pareto distribution. Consequently, the cloud servers will receive a number of requests that require high service times which leads to having an increase in the overall cloud response time.

The average resource utilization evaluation for simulation case 2 [RR-U-P] is shown in Figure 6.4. It is noted that there is an increase in the resources utilization while using the Pareto distributions as an input traffic when comparing the exponential distributions results shown in Figure 6.2 to the Pareto distributions results shown in Figure 6.4. The increase is also due to the heavy right tail of the Pareto distribution.

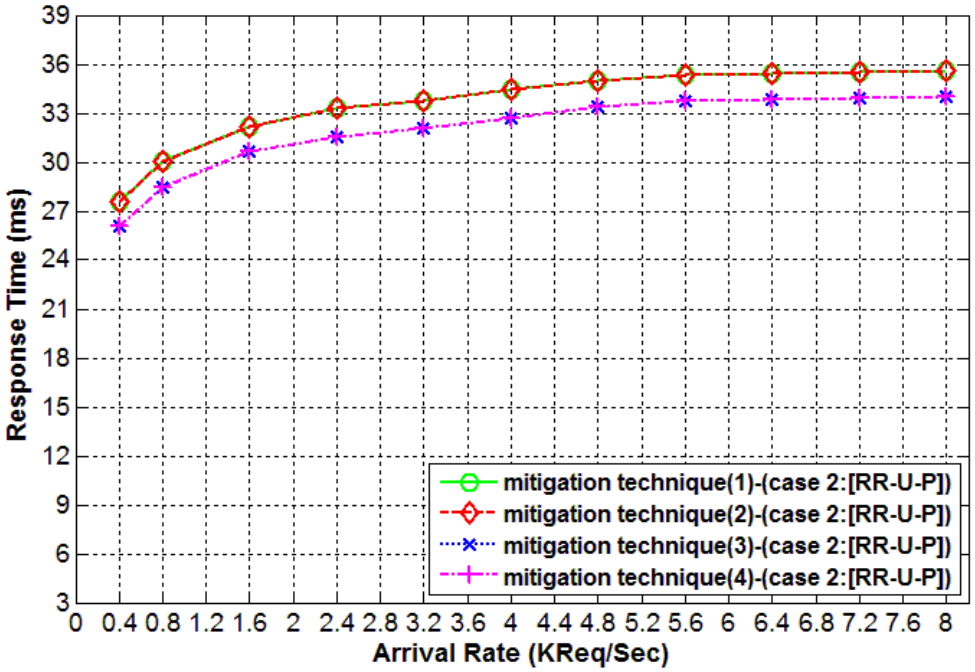


Figure 6.3: Response time result in the normal mode after applying case 2 [RR-U-P].

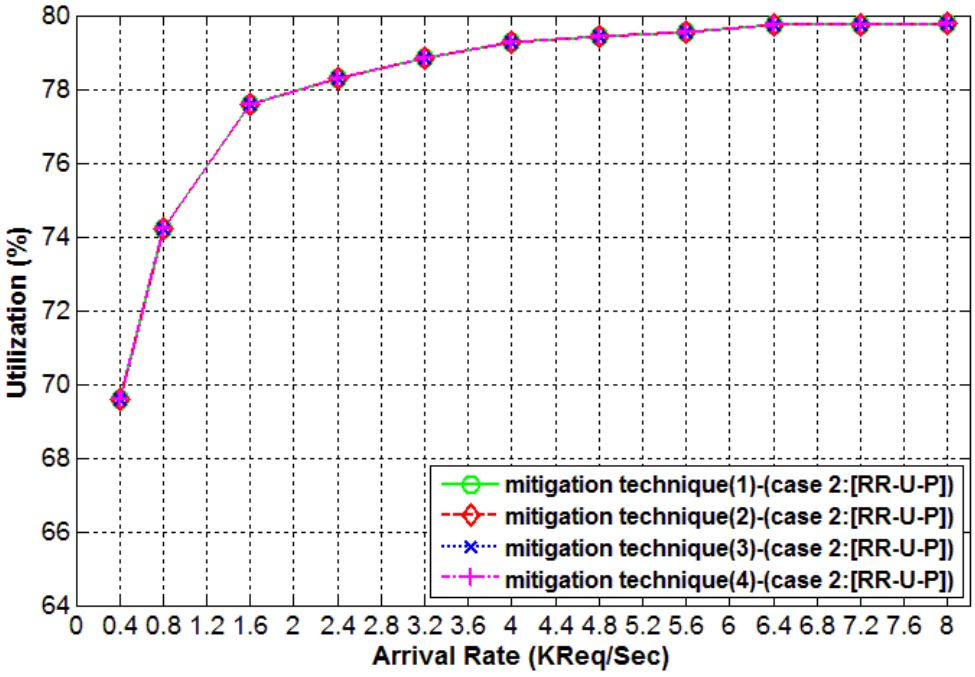


Figure 6.4: Resource Utilization result in the normal mode after applying case 2 [RR-U-P].

6.1.3 Response Time and Utilization Results for Simulation case 3 [LL-U-E]

Figure 6.5 shows the cloud response time evaluation for different arrival rates while using the parameters specified in case 3 [LL-U-E]. Comparing the results of Figure 6.1 and Figure 6.5, it is clear that the LL dispatching algorithm outperforms the RR algorithm in all mitigation techniques in term of response time. This is due to fact that the LL algorithm utilizes the cloud resources more efficiently than the RR algorithm. For example in mitigation technique (1) with an arrival rate equal to 400 Req./Sec. the RR algorithm is approximately 2.75 times worse than the LL algorithm in terms of the cloud response time. However, as the arrival rate increases the performance of the RR algorithm becomes closer to the LL algorithm as all the VM instances in the cloud will be highly utilized. Hence, the equal distribution of the requests among the servers in the RR algorithm becomes almost the same as choosing the least loaded server in the LL algorithm.

The average resources utilization results for the four mitigation techniques under study after applying case 3 [LL-U-E] parameters are shown in Figure 6.6. Comparing the results of Figure 6.2 and Figure 6.6, it is evident that the average utilization result improves when the LL algorithm is used. At the first glance, the increase in the average utilization can be attributed to the asymmetrical distribution of requests among the servers while using the LL algorithm. However, this is not true since this should lead to a different average utilization in each cloud server but not to the overall utilization of all servers of the cloud as explained in the next paragraph. The real reason for this increase in the average utilization can be explained by considering equation 4.2. Due to the LL algorithm, the finishing time of each server will be less. Hence, according to equation 4.2, the overall average utilization will increase.

Figure 6.7 and Figure 6.8 show a comparison between the number of requests served by the cloud servers in mitigation technique (1) within one minute of simulation at 400 Req./Sec. while using simulation case 1 [RR-U-E] and simulation case 3 [LL-U-E], respectively. It is clear that the equal distribution of requests among servers when using the RR algorithm and the asymmetrical distribution of requests among servers when using the LL algorithm. Similar figures will result for the other mitigation techniques as in Figure 6.7 and Figure 6.8.

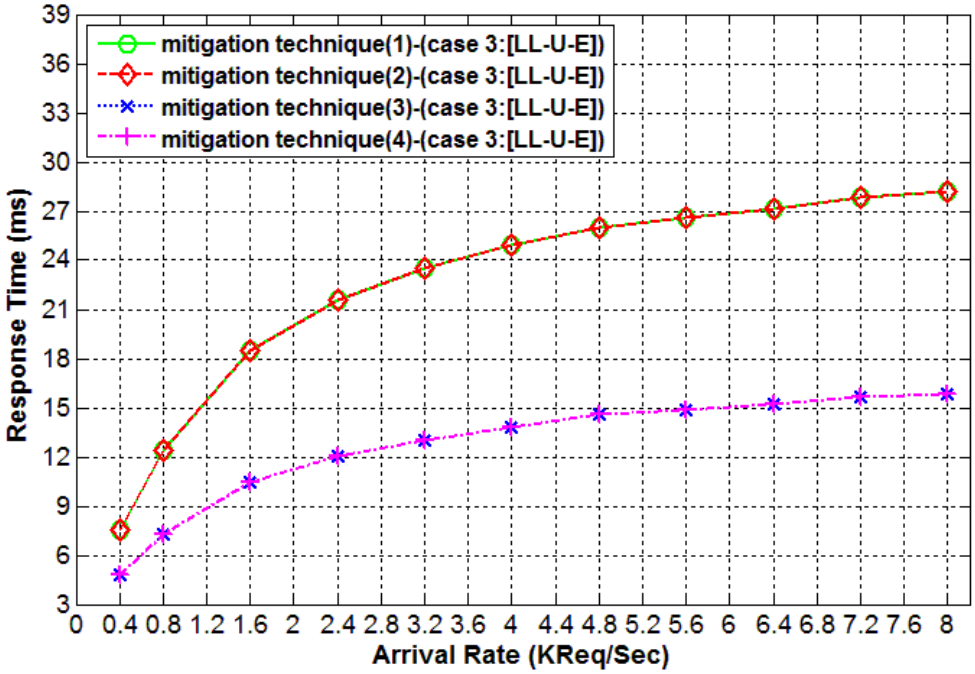


Figure 6.5: Response time result in the normal mode after applying case 3 [LL-U-E].

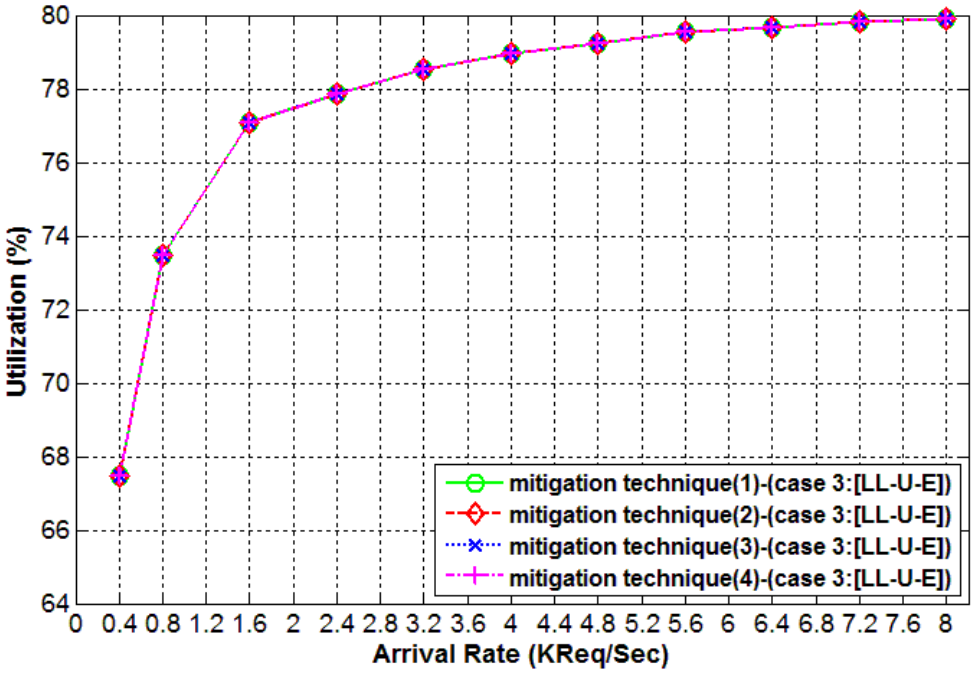


Figure 6.6: Resource Utilization result in the normal mode after applying case 3 [LL-U-E].

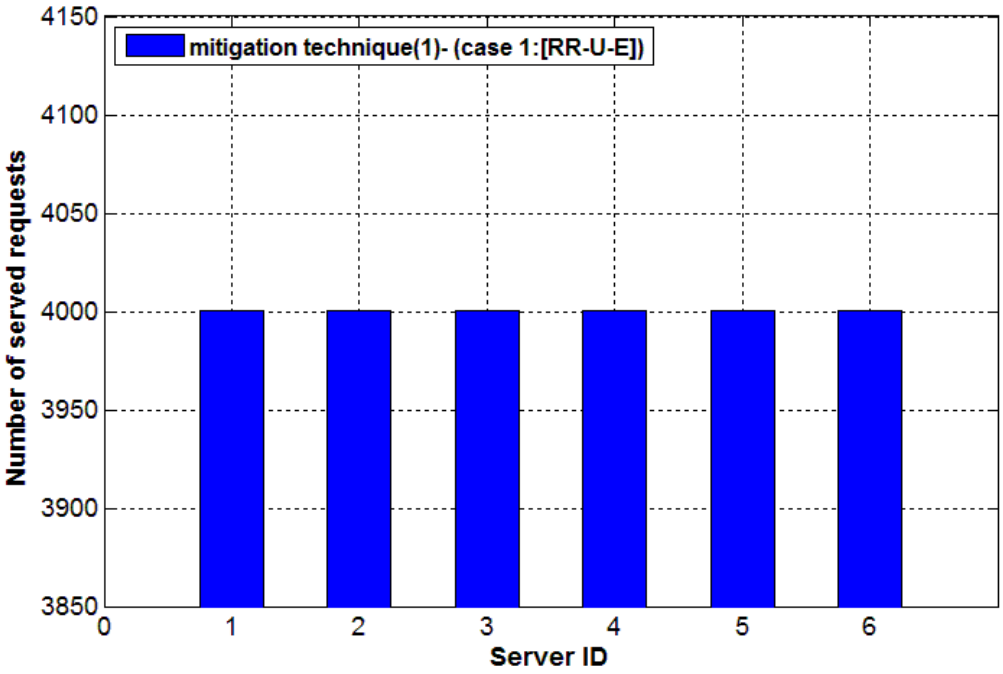


Figure 6.7: The number of requests served in each server while using the case 1[RR-U-E] parameters within one minute of simulation at 400 Req./Sec. for mitigation technique (1).

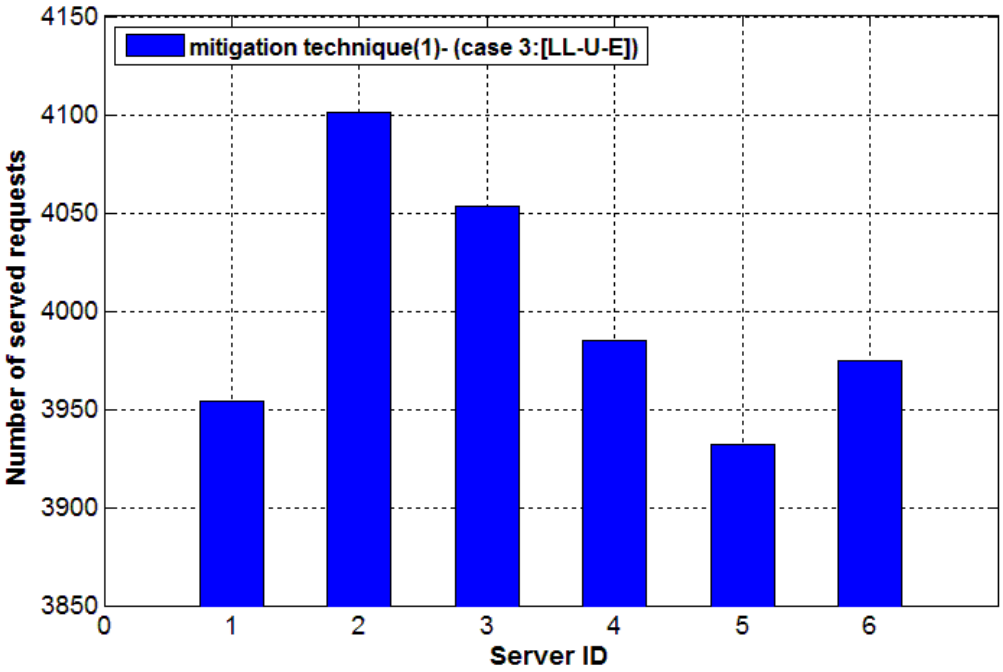


Figure 6.8: The number of requests served in each server while using case 3 [LL-U-E] parameters within one minute of simulation at 400 Req./Sec. for mitigation technique (1).

6.1.4 Response Time and Utilization Results for Simulation case 4 [LL-U-P]

The average response time results and the average utilization results for the four mitigation techniques under study after applying the parameters of case 4 [LL-U-P] are depicted in Figure 6.9 and Figure 6.10, respectively. Comparing the results of Figure 6.5 and Figure 6.9, it is clear that there is an increase in the cloud response time for case 4. Similarly, by comparing Figure 6.6 and Figure 6.10, it is noted that there is an increase in the cloud resources utilization for case 4. The reasons behind both observations are similar to those that were discussed in section 6.1.2. Moreover, mitigation techniques (3) and (4) provide a significant decrease in the overhead as compared to mitigation techniques (1) and (2) as the vFirewall of mitigation techniques (1) and (2) checks the IP addresses of incoming requests and sends Turing tests to classify the users in whitelist and blacklist.

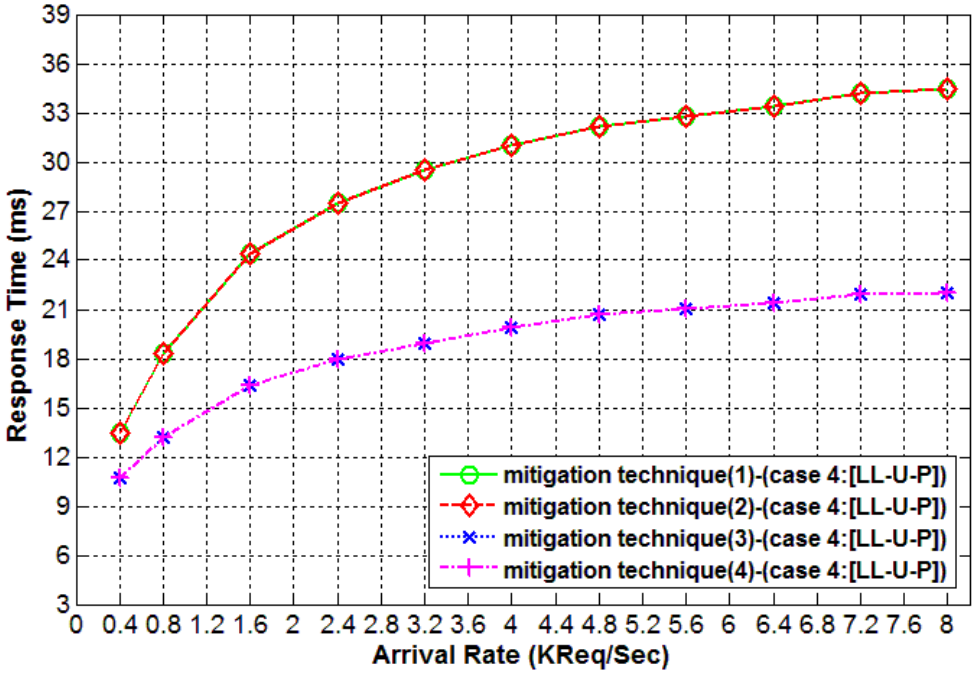


Figure 6.9: Response time result in the normal mode after applying case 4 [LL-U-P].

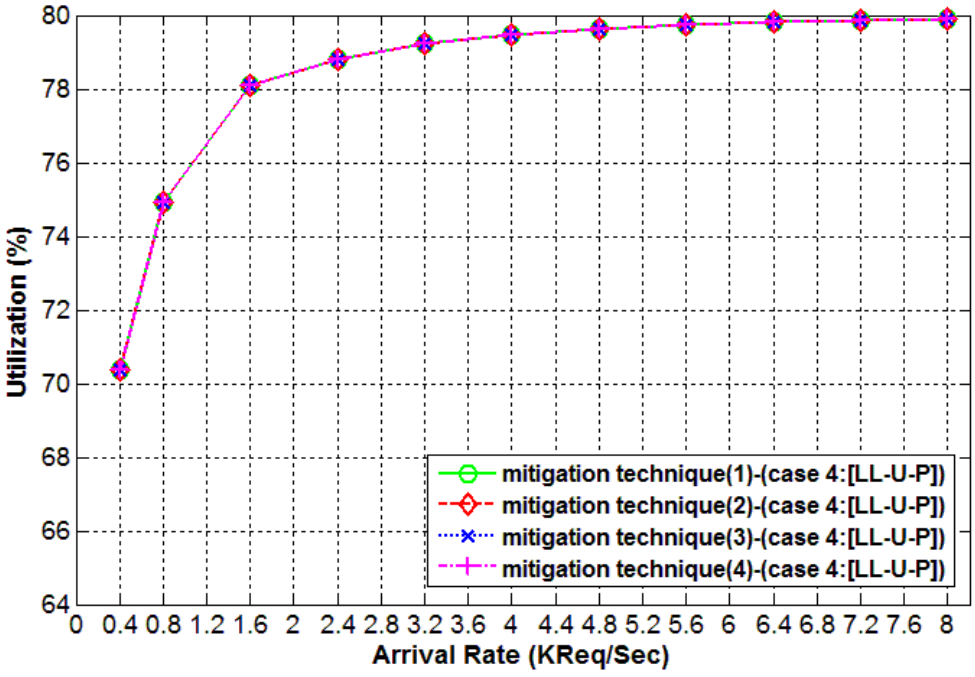


Figure 6.10: Resources Utilization result in the normal mode after applying case 4 [LL-U-P].

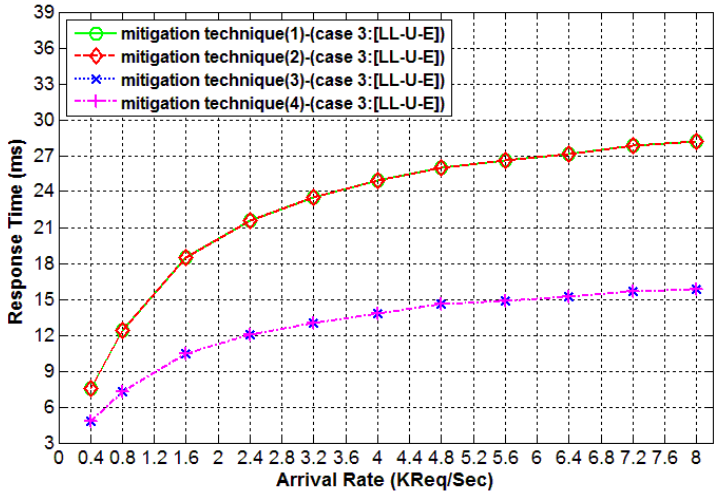
6.1.5 Comparison Between Simulations Results of case 3 [LL-U-E] and case 5 [RR-CAP-E]

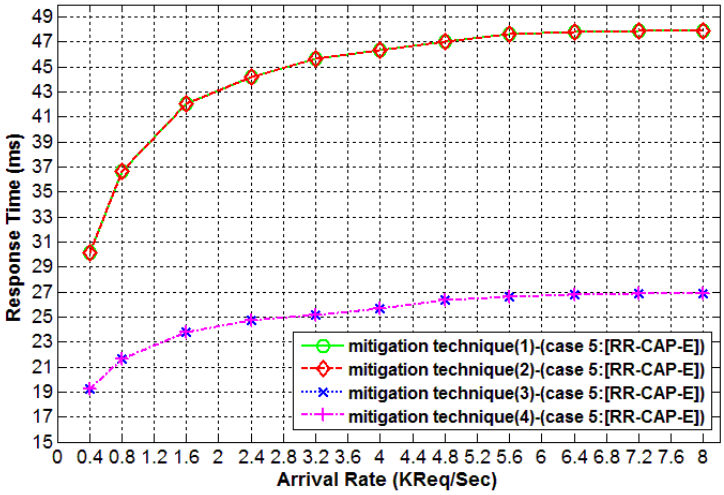
Section 6.1.1 through section 6.1.4 demonstrated that the simulation results of case 3 [LL-U-E] outperform the simulation results of the other cases in terms of response time as well as utilization. Therefore, a comparison between the usage of case 3 parameters and case 5 parameters are presented in this section. The purpose of this comparison is to find out the amount of system performance improvement after applying case 3 parameters as compared to the original simulation results reported by the authors of these techniques.

Figure 6.11 shows the cloud response time evaluation for different arrival rates while using the parameters specified in case 3 [LL-U-E] and case 5 [RR-CAP-E]. The gap between the cloud response time of mitigation techniques (1) and (2) and the cloud response time of mitigation techniques (3) and (4) in case 3 is smaller than the corresponding gap in case 5. The reason behind this is related to the large relative difference between the average response time in solving CAPTCHA Turing test and the average response time in responding to a URL redirection. Moreover, from the normalized response time shown in Figure 6.11, it can be concluded that by using the parameters of case 3 the cloud response time in mitigation technique (1) is at most 4 times better than in case 5 [RR-CAP-E]. The reason behind this improvement is due to the usage of the LL algorithm and the URL redirection technique instead of the RR algorithm and the CAPTCHA Turing test. Moreover, as the arrival rate increases the improvement starts to decay since at high rates all the VM instances in the cloud will be highly utilized. Hence, the equal distribution of the requests among the servers in the RR algorithm becomes almost the same as choosing the least loaded server in the LL algorithm. At high rates, the factor that plays the major

role in the system improvement becomes the usage of the URL redirection technique instead of the CAPTCHA Turing test.

The average resources utilization results for the four mitigation technique under study after applying case 3 [LL-U-E] parameters and case 5 [RR-CAP-E] parameters are shown in Figure 6.12. From the normalized utilization, it is clear that a slight increase in the average cloud resources utilization is gained as a result of applying case 3 parameters on the four mitigation techniques.





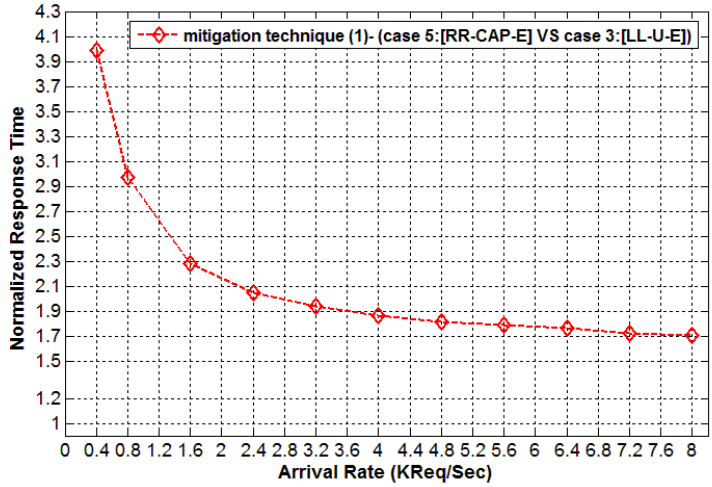


Figure 6.11: Response time comparison between case 3 [LL-U-E] and case 5 [RR-CAP-E] in the normal mode.

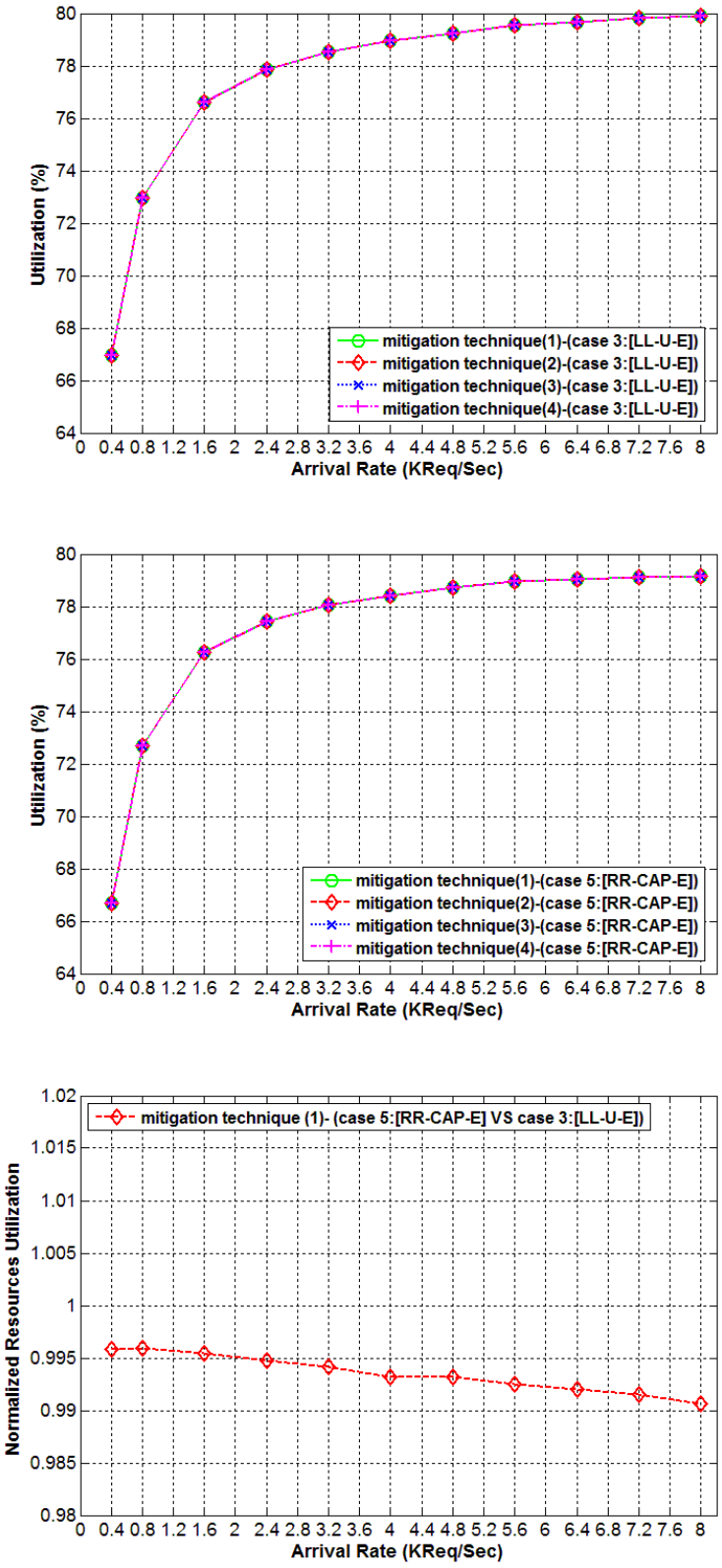


Figure 6.12: Resources Utilization comparison between case 3 [LL-U-E] and case 5 [RR-CAP-E] in the normal mode.

6.2 Flash Overcrowd Mode Results

The flash overcrowd phenomena occurs when an enormous amount of legitimate traffic arrives at the cloud in a short period of time. A successful EDoS mitigation technique should be able to rapidly detect and differentiate between such a phenomena and an attacker's behavior so as not to impact legitimate users traffic. Accordingly, this mode checks if the mitigation techniques under study are capable of handling the flash overcrowd load without adding overhead to the cloud service. In addition, this mode evaluates the efficiency of the auto scaling in each of the four mitigation techniques under study when the cloud service faces a flash traffic.

In order to simulate an experiment that mimics the flash overcrowd behavior, the cloud system are set to receive normal traffic of 400 Req./Sec. for the first 5 minutes of simulation, and then the traffic increases until it reaches 2000 Req./Sec. at minute six. After that, the load remains at 2000 Req./Sec. until the end of the simulation. Hence, the cloud service encounters a high load peak that is 5 times larger than the normal traffic. The aforementioned flash overcrowd load is depicted in Figure 6.13. Based on equation (4.8) the number of initial running VMs to handle the 400 Req./Sec. without exceeding the utilization upper threshold is set to 6 VMs. Also, attack traffic is not considered during the flash overcrowd mode.

Figure 6.14 shows the evaluation of the auto-scaling mechanism in the studied mitigation techniques when the cloud service encounters a flash overcrowd load. In mitigation techniques (1) and (2) the average CPU utilization of the cloud instances is periodically checked every 5 minutes. Subsequently, if the average CPU utilization exceeds the upper threshold of 80%, two additional instances will be allocated after a provisioning period of

55.4 seconds. On the other hand, mitigation techniques (3) and (4) check the average CPU utilization of the cloud instances every 1 minute. Accordingly, if the CPU utilization threshold is crossed, an additional instance is added to the running instances after a one minute of provisioning overhead. For example, at minute six of the simulation, mitigation techniques (3) and (4) check the CPU utilization of the cloud and request an addition of one VM to the running set. The additional instance starts working at minute seven. Mitigation techniques (1) and (2) check the utilization at minute five and find the CPU utilization threshold is still uncrossed, after that at minute ten mitigation techniques (1) and (2) check the CPU utilization again and identify that the CPU utilization has been crossed. So, two additional instances will join the running instances at second 10.54. It can be concluded from Figure 6.14 that mitigation techniques (3) and (4) converged to the required number of instances faster than mitigation techniques (1) and (2).

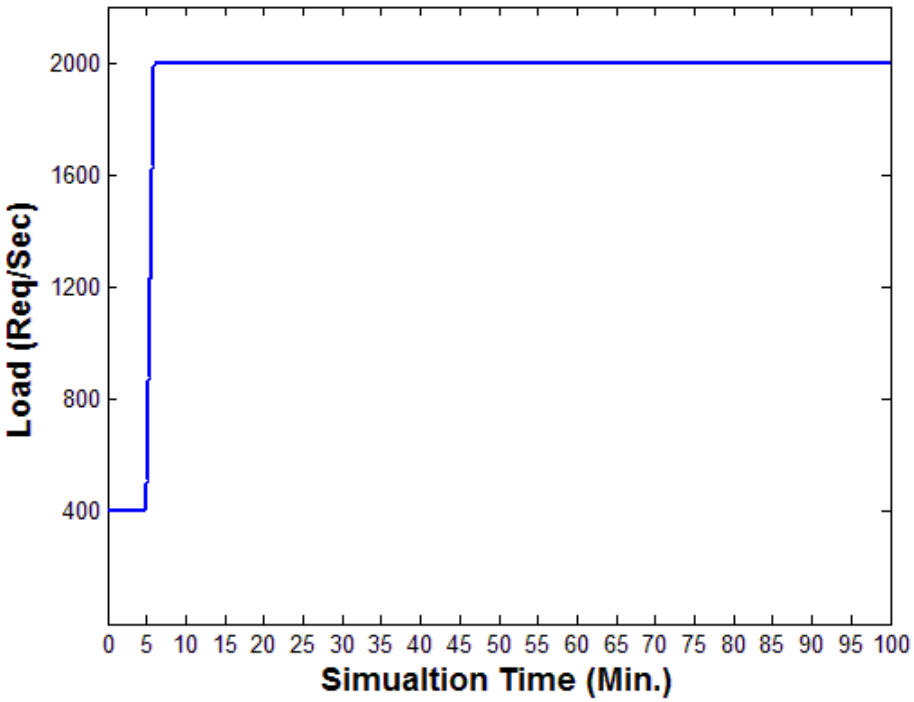


Figure 6.13: Flash overcrowd traffic.

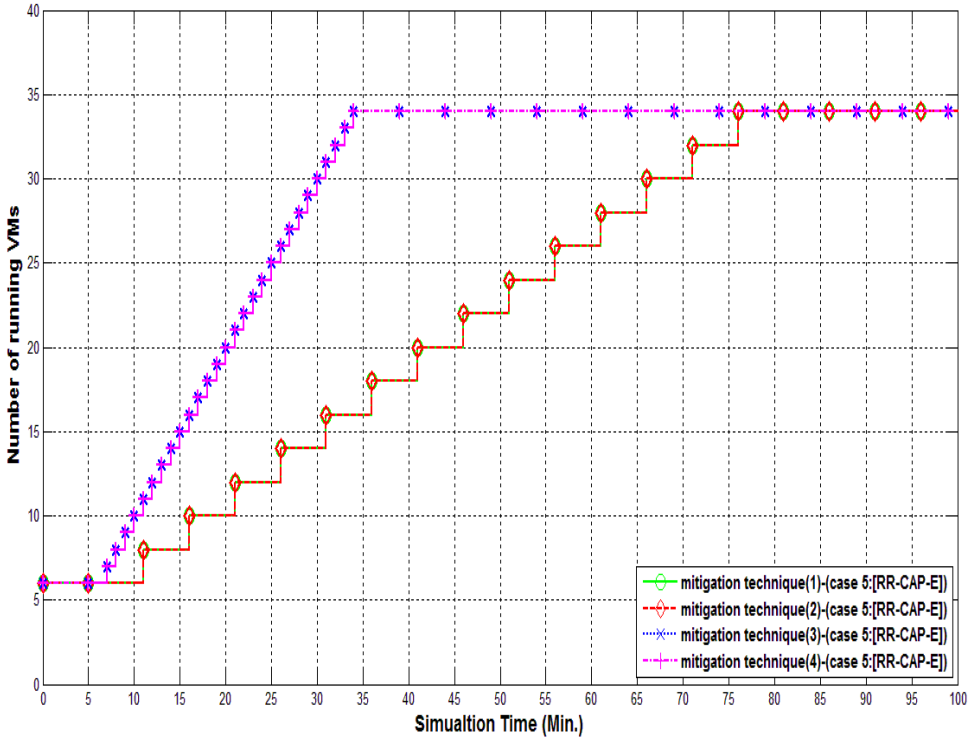


Figure 6.14: Simulation results of the number of allocated VMs at a flash overcrowd rate of 2 KReq./Sec.

6.2.1 Comparison Between Simulation Results of case 3 [LL-U-E] and case 5 [RR-CAP-E]

A comparison between the required number of VMs needed to serve the flash overcrowd users while using the four mitigation techniques and the analytical method discussed in equation (4.8) is shown in Figure 6.15. The Figure shows that when the arriving traffic is equal to 400 Req./Sec. the required instances while using either the analytical method or the mitigation techniques under study is the same since there is no auto-scaling needed in this situation. On the contrary, when the cloud receives traffic that is larger than or equal 800 Req./Sec. all the mitigation techniques allocate more cloud instances than the analytical method. This is regards to the accumulated queuing requests in the initial running VMs that operate from the beginning of the simulation. Accordingly, the accumulated queuing requests make the average utilization in the cloud above 80%. As consequence, the auto scaling feature will append more VMs to the running instances set.

The response time results in the flash overcrowd mode after applying case 5 [RR-CAP-E] are depicted in Figure 6.16. Note that there is no special algorithm in the four mitigation techniques under study capable of distinguishing between the flash overcrowd (FC) traffic and the attack traffic. Consequently, all those techniques send one or more CAPTCHA Turing tests towards the cloud users that generate the FC traffic. Furthermore, the overhead associated with each mitigation technique will be different from one technique to another. For mitigation techniques (1) and (2) where a whitelist table is used in the vFirewall, the cloud user that generates the FC traffic is forced to solve only one Turing test to gain the full access to the cloud resources. On the other hand, for mitigation technique (3) the cloud user is forced to solve more than one CAPTCHA Turing test according to the rate limit algorithm. Because the VMs utilization threshold will be crossed as a result of the flash

traffic, so the VM observer will order the firewall to send all the subsequent requests to the VM investigator. In turn, the VM investigator sends an additional CAPTCHA to each user in order to delay the incoming traffic until the cloud utilization decreases. In mitigation technique (4), the cloud user that generates FC traffic is also forced to solve CAPTCHA Turing test whenever the average resources CPU utilization is crossed. Therefore, the user in mitigation techniques (3) and (4) will suffer from relatively high cloud response time. Also, Figure 6.16 shows that when the arriving traffic is equal to 400 Req./Sec. the cloud response time is higher than the associated cloud response time when the cloud receives traffic that is larger than or equal 800 Req./Sec. for all the mitigation techniques. This dip is regard to the usage of the initial number of running VMs while the arriving traffic is equal to 400 Req./Sec. without needing to auto scale which makes these VMs highly utilized. Accordingly, the cloud response time will increase. On the contrary, when the arriving traffic is larger than or equal to 800 Req./Sec. all mitigation technique append more cloud instances to the running instances set which resulted in decreasing the queuing delay in each server and accordingly the cloud response time.

Figure 6.17 shows the cloud response time results in the flash overcrowd mode after applying case 3 [LL-U-E] parameters. There is a noticeable reduction in the cloud response time for all the studied mitigation techniques when comparing the results provided in Figure 6.16 and the results provided in Figure 6.17. The reason behind this is related to the large relative difference between the average response time in solving CAPTCHA Turing test and the average response time in responding to a URL redirection as well as using the LL algorithm instead of RR algorithm.

Similarly, the resource utilization results in the flash overcrowd mode after applying case 5 [RR-CAP-E] parameters is depicted in Figure 6.18. When the resource utilization results shown in Figure 6.2 are compared with and the resource utilization results provided in Figure 6.18, we see a decrease in the resources utilization in the flash overcrowd mode. The reason behind this decrease is due to using additional cloud instances in the flash overcrowd mode than in the case of the normal mode. Also, Figure 6.18 shows that when the arriving traffic is equal to 400 Req./Sec. the resource utilization results is higher than the associated resource utilization results when the cloud receives traffic that is larger than or equal 800 Req./Sec. for all the mitigation techniques. The reason behind this is previously discussed while addressing the same phenomena in Figure 6.16.

Finally, Figure 6.19 shows the resource utilization results in the flash overcrowd mode after applying case 3 [LL-U-E] parameters. It is clear that a slight increase in the average cloud resources utilization is gained as a result of applying case 3 parameters to the four mitigation techniques. The reason behind this increase is the same as that presented in section 6.1.3.

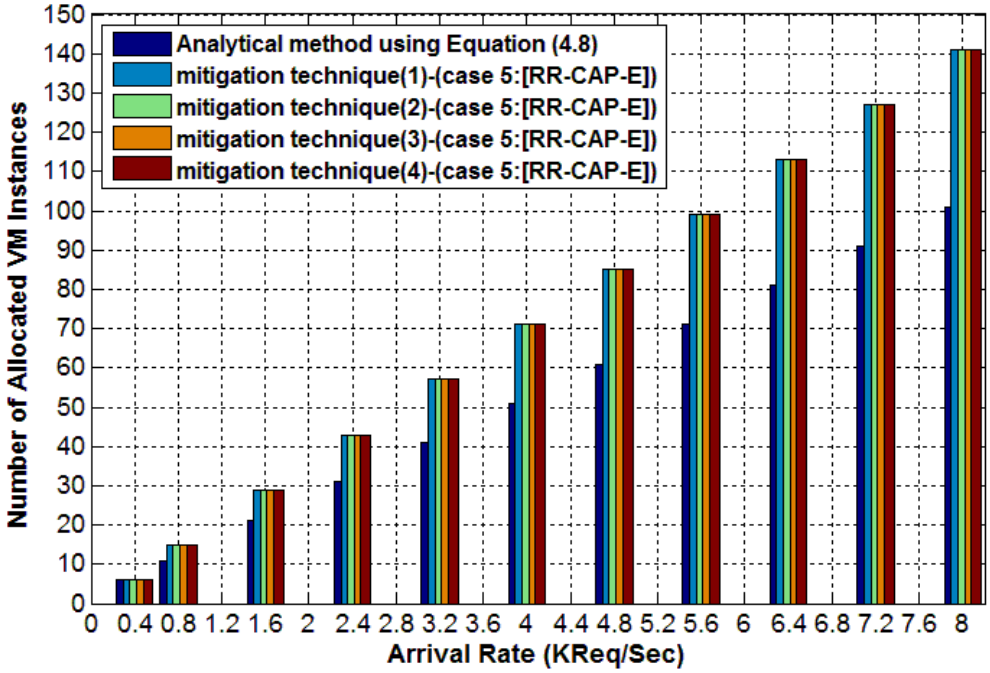


Figure 6.15: Simulation results of the number of allocated VMs at different flash overcrowd rates.

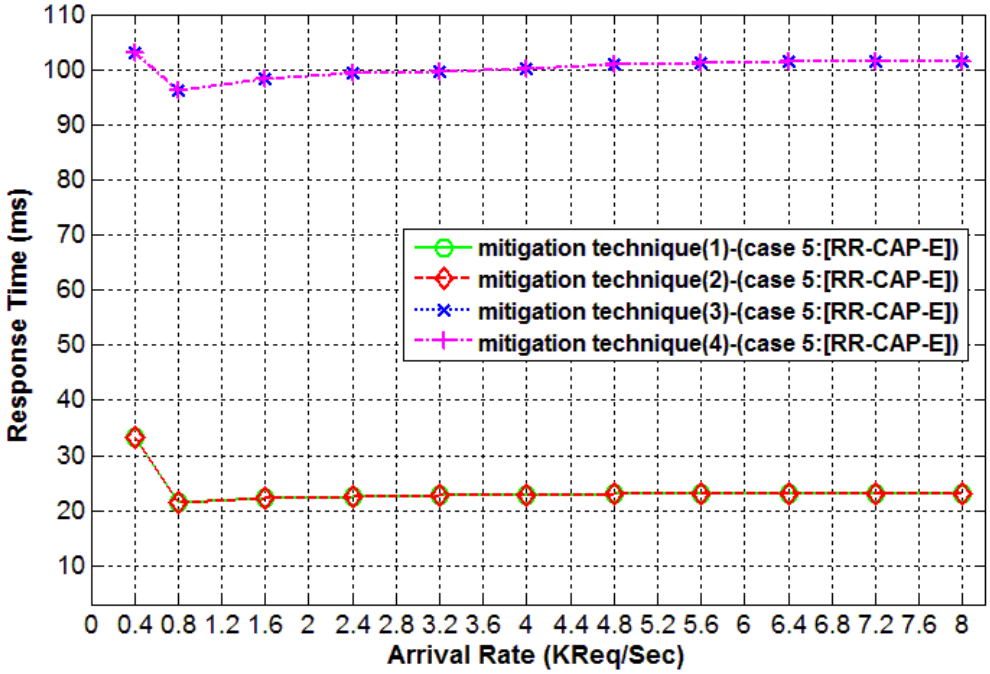


Figure 6.16: Response time results in the flash overcrowd mode after applying case 5 [RR-CAP-E].

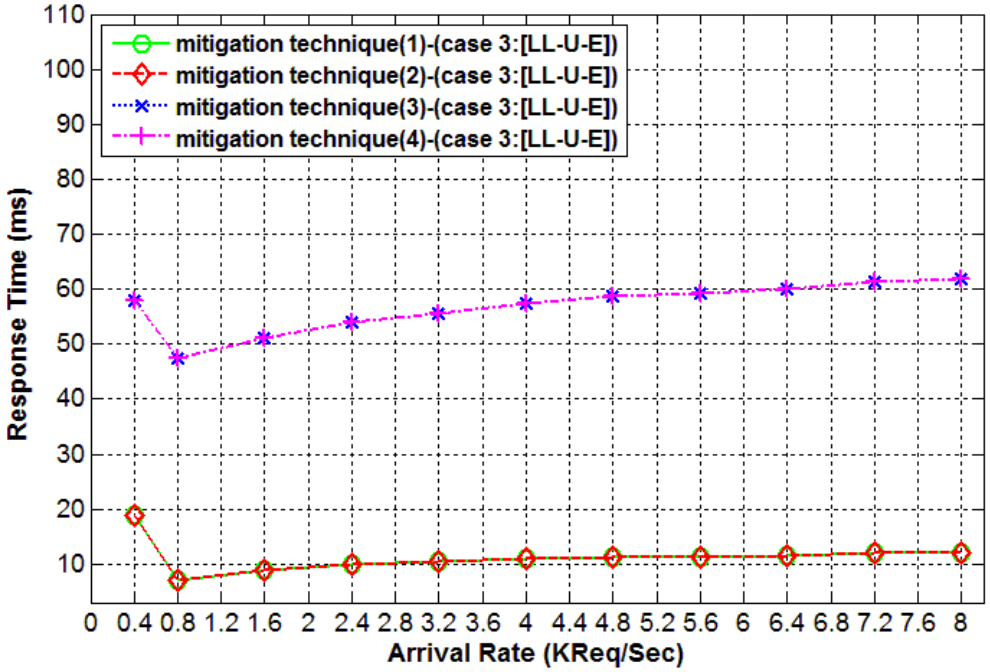


Figure 6.17: Response time results in the flash overcrowd mode after applying case 3 [LL-U-E].

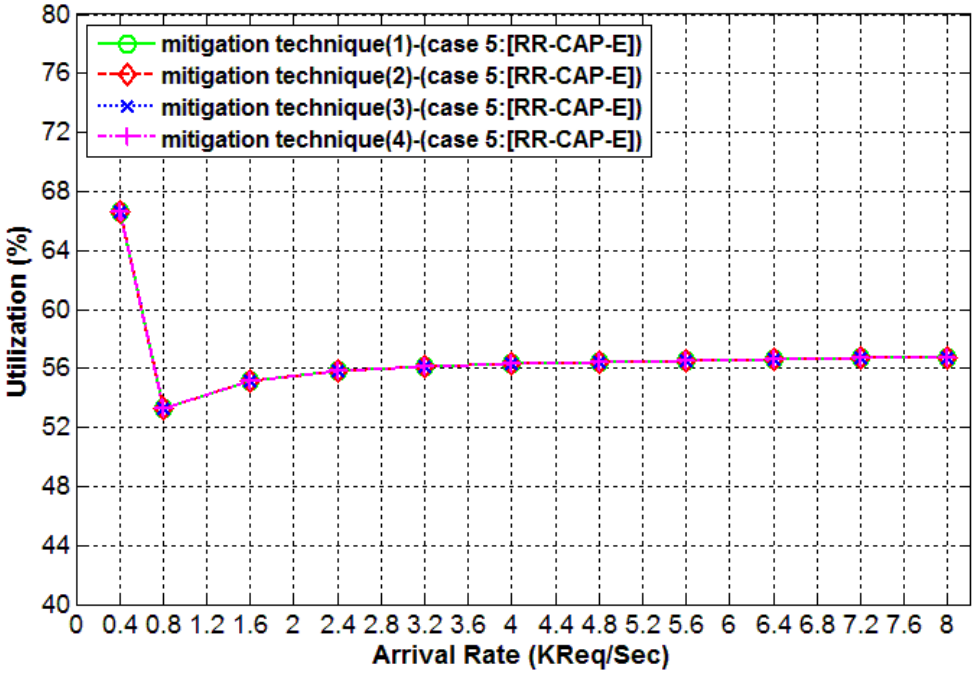


Figure 6.18: Resources Utilization results in the flash overcrowd mode after applying case 5 [RR-CAP-E].

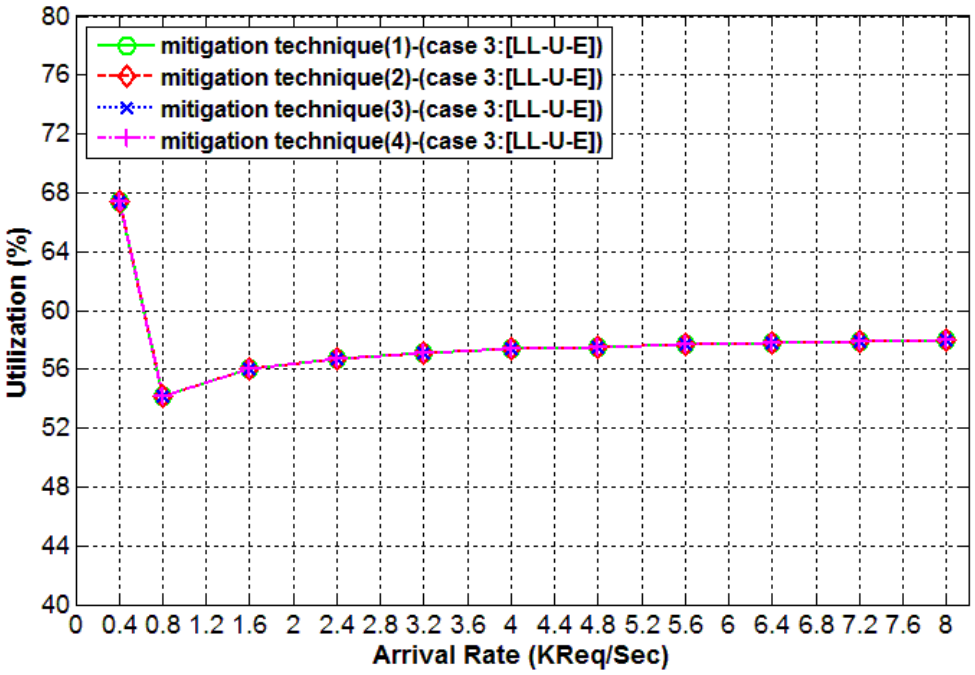


Figure 6.19: Resources Utilization results in the flash overcrowd mode after applying case 3 [LL-U-E].

6.3 Attack Mode Results–IP spoofing

In order to evaluate the effect of the EDoS attack on the four mitigation technique under study, the cloud service is tested using a fixed load of legitimate traffic that equals to 400 Req./Sec. and to a variable attack traffic that is varied between 400 and 8000 Req./Sec. For every attack rate, the number of initial running instances in the studied mitigation technique is set to 6 VMs as deduced from the results that shown in Figure 6.15.

In the attack mode, all EDoS mitigation techniques are susceptible to the IP Spoofing problem. Thus, the four considered mitigation techniques are studied under two cases in order to cover the spoofing problem: the whitelist case and the blacklist case. In the whitelist case, the IP address of the attacker's machine is placed in the whitelist by sending one legitimate request from the attacker's machine. Then, the attacker orders a set of bot machines to generate a huge amount of attacking traffic toward the cloud while setting their IP addresses to the attacker's whitelisted IP address. On the other hand, the blacklist case occurs when initially the mitigation technique identifies a number of spoofed attackers IP addresses and places these addresses in the blacklist. A problem might occur when a legitimate user sends a request towards the cloud service using an IP address that is already blacklisted. Consequently, the mitigation technique will drop that request and block the legitimate user from being served.

6.3.1 Comparison Between Simulation Results of case 3 [LL-U-E] and case 5 [RR-CAP-E] (Whitelist case)

The required number of VMs needed to serve the cloud users in the attack mode (whitelist case) while using the four mitigation techniques is shown in Figure 6.20. Both mitigation techniques (1) and (2) have allocated more VMs instances than mitigation techniques (3) and (4) as mitigation techniques consider the attacking traffic as a flash traffic. More specifically, all of the attack traffic in mitigation technique (1) is considered as a flash traffic. Similarly, only the attack traffic that have same IP address and TTL value of a whitelisted user is served and considered as a flash traffic in mitigation technique (2). As a result, both mitigation techniques (1) and (2) will auto scale to serve the flash traffic. On the other hand, mitigation techniques (3) and (4) are considered reactive mitigation techniques and will start working when the average CPU utilization of the cloud resources exceeds the threshold. Hence, some of the attack traffic will be served until the utilization threshold is crossed. Afterwards, all the requests in the vFirewall will be sent to the VM investigator for further verification and a onetime auto scaling that adds one additional VM is performed.

The response time results in the attack mode (whitelist case) after applying case 5 [RR-CAP-E] parameters are depicted in Figure 6.21. There is an increase in the cloud response time in mitigation technique (1) and (2) when the attack traffic increases. This is mainly due to the fact that all of the attack traffic is served by the cloud in mitigation technique (1), and some of the attack traffic is served in mitigation technique (2) as explained earlier. In mitigation techniques (3) and (4) the response time results are constant. Accordingly, when these mitigation techniques are in operation they will eliminate all of the attack traffic successfully.

Figure 6.22 shows the response time results in the attack mode (whitelist case) after applying case 3 [LL-U-E] parameters. There is a noticeable reduction in the cloud response time for all the studied mitigation techniques as compared to the results provided in Figure 6.21. The reason behind this is related to the large relative difference between the average response time in solving a CAPTCHA Turing test and the average response time in responding to a URL redirection as well as using LL algorithm instead of RR algorithm.

The resource utilization results in the attack mode (whitelist case) after applying case 5 [RR-CAP-E] parameters are depicted in Figure 6.23. In mitigation techniques (3) and (4), the CPU utilization results are similar because both mitigation techniques employ the same number of VM instances to serve the cloud users. The CPU utilization result in mitigation technique (1) is greater than the CPU utilization in mitigation technique (2) since less number of attack requests are considered as flash traffic in mitigation technique (2). Comparing Figure 6.23 to Figure 6.24 reveals a slight increase in the average cloud resources utilization for case 3 as a result of applying case 3 parameters to the four mitigation techniques. The reason behind this increase was explained in section 6.1.3.

Figure 6.25 shows the number of False Negative requests served by the cloud resources for the whitelist case. The entire attack traffic is severed by the cloud resources in mitigation technique (1) due to considering these requests as flash requests. In mitigation technique (2) only the attack traffic that has the same IP address and TTL value of a whitelisted user is served and considered as flash traffic. Accordingly, the behavior observed in Figure 6.25 reflects the assumption made in [6] of having 20% of the attack traffic having the same IP address and TTL value of a whitelisted users. Mitigation techniques (3) and (4) will start working when the average CPU utilization of the cloud resources exceeds the threshold.

Hence, some of the attack traffic will be served until the utilization threshold is crossed. Afterwards, all the requests in the vFirewall will be sent to the VM investigator for further verification and both mitigation techniques will successfully identify and drop all the attack traffic.

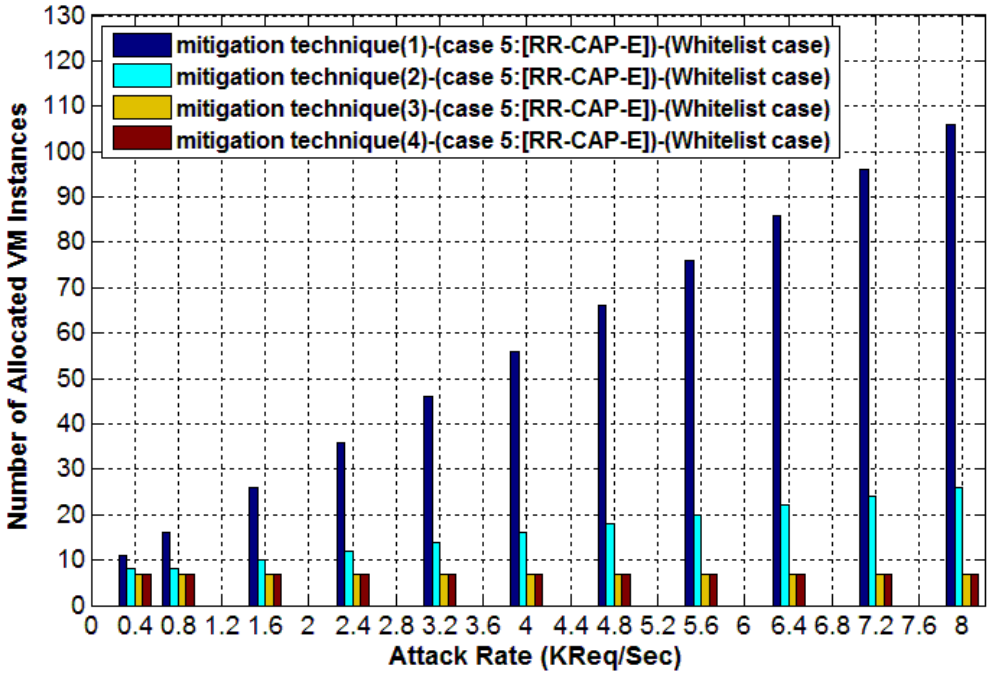


Figure 6.20: Simulation results of the number of allocated VMs at different attack rates for the Whitelist case.

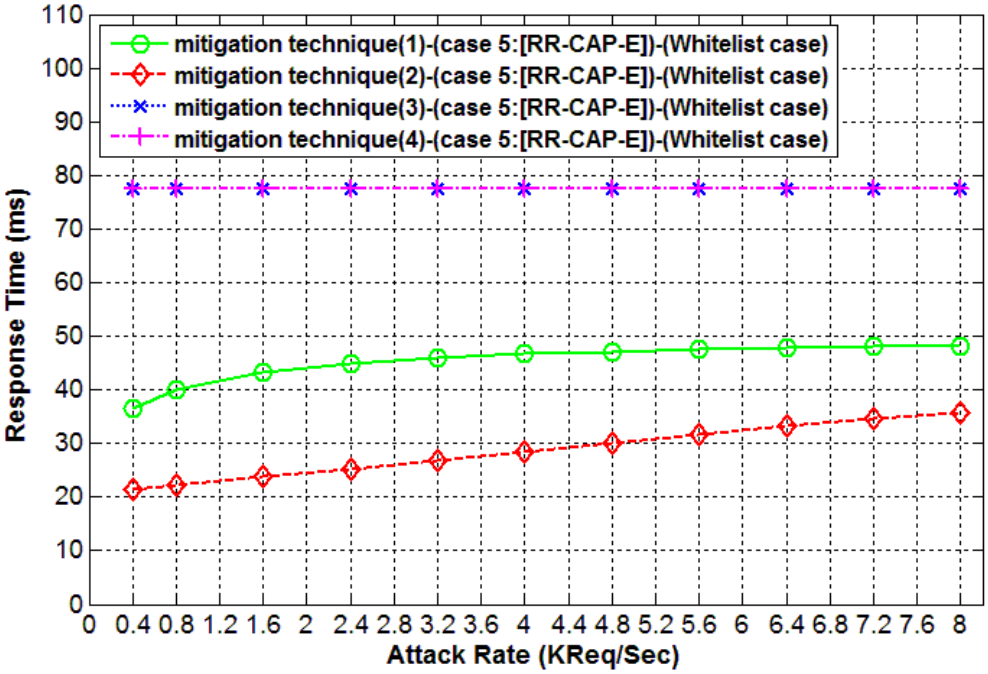


Figure 6.21: Response time results in the attack mode after applying case 5 [RR-CAP-E] for the whitelist case.

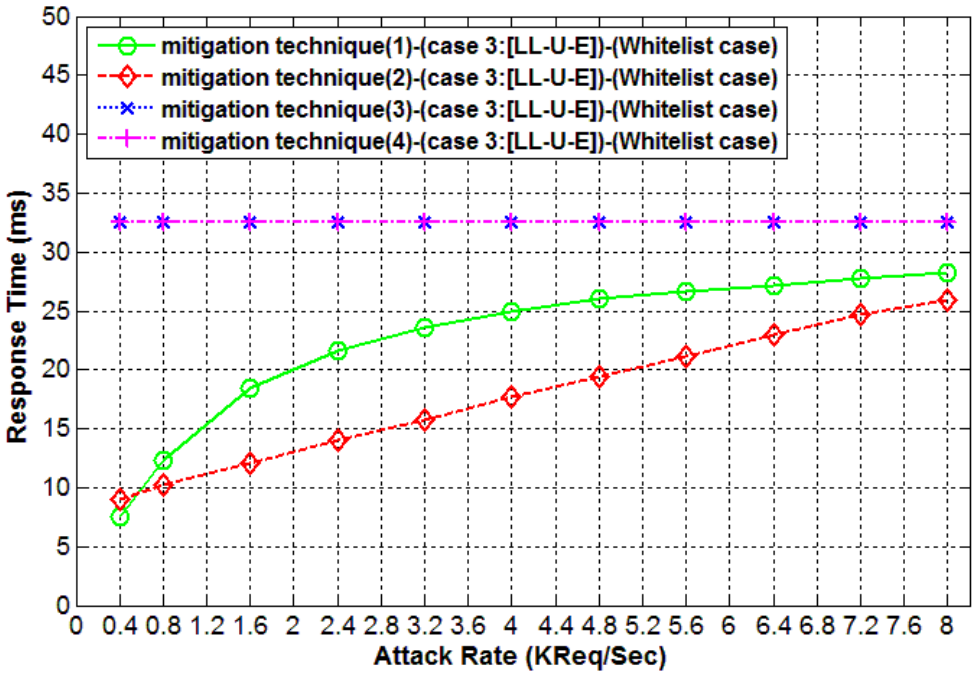


Figure 6.22: Response time results in the attack mode after applying case 3 [LL-U-E] for the whitelist case.

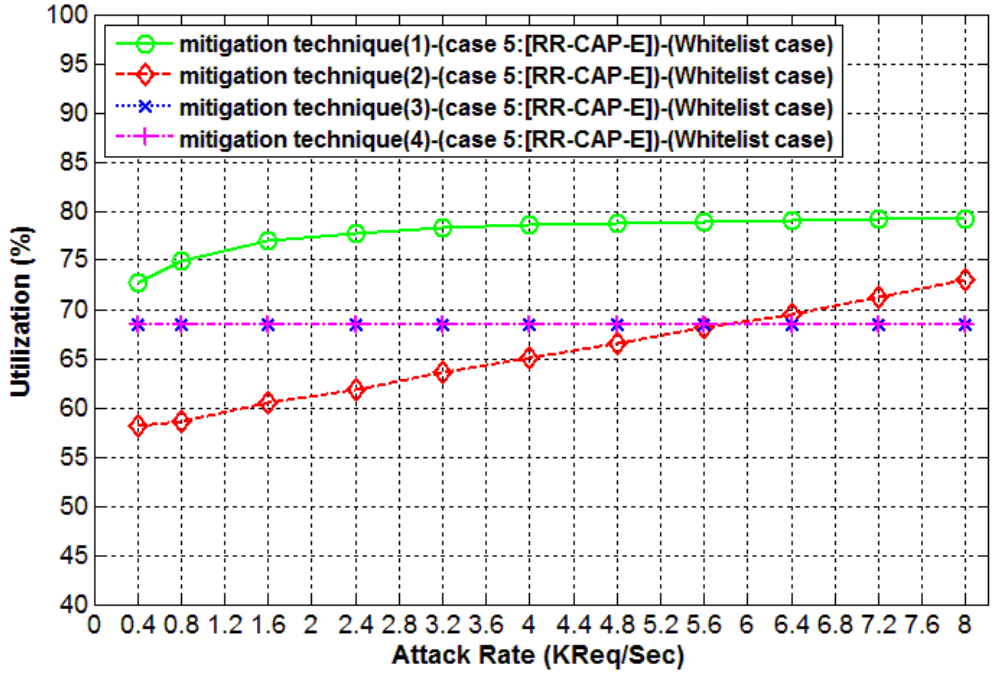


Figure 6.23: Utilization results in the attack mode after applying case 5 [RR-CAP-E] for the whitelist case.

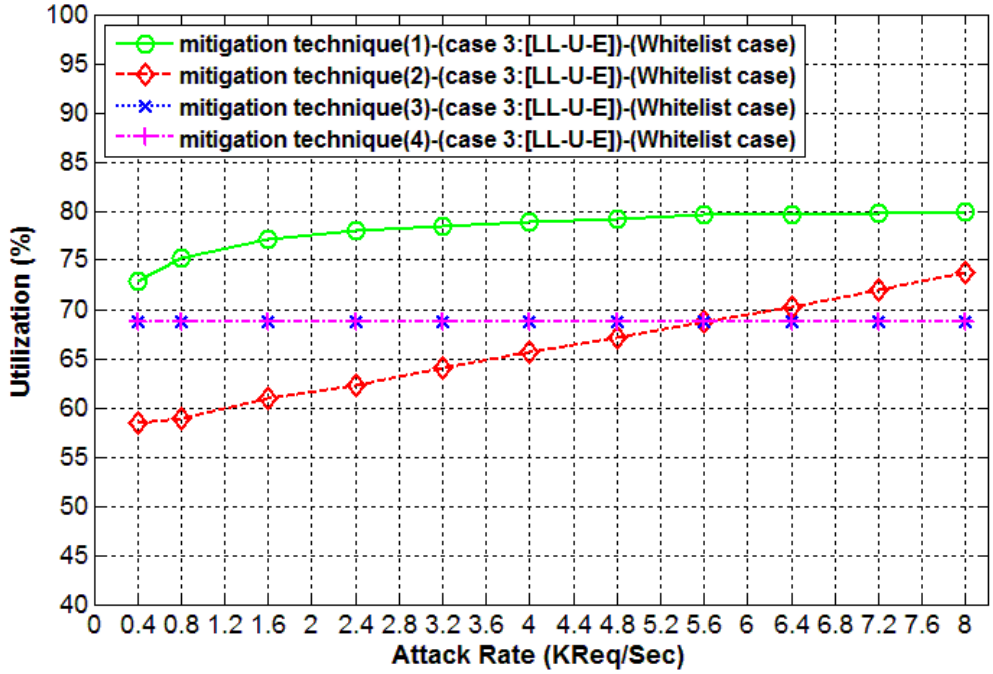


Figure 6.24: Utilization results in the attack mode after applying case 3 [LL-U-E] for the whitelist case.

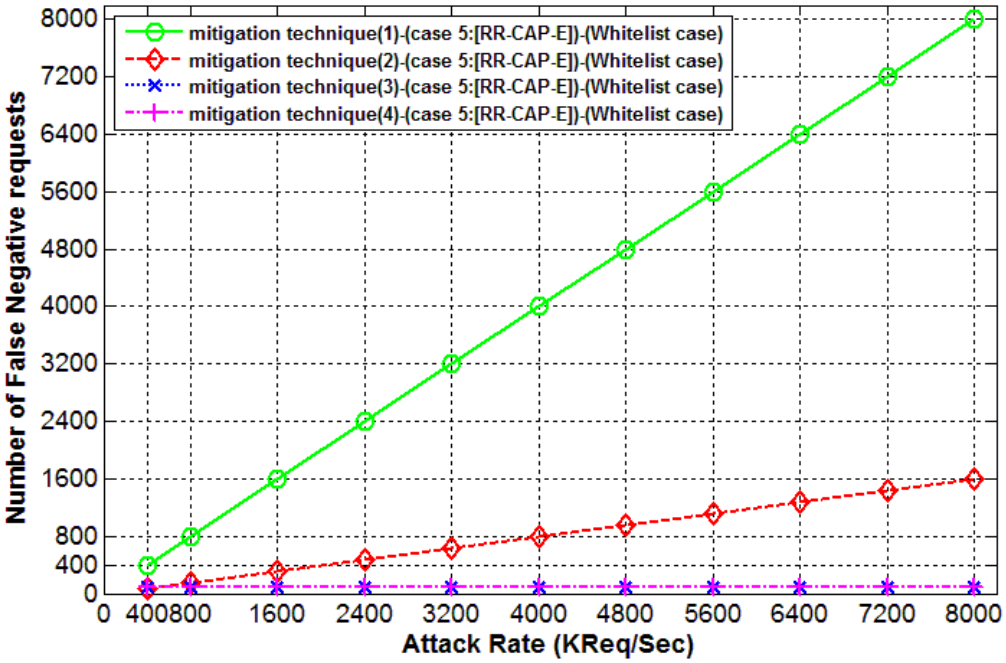


Figure 6.25: The number of False Negative requests in the attack mode for the whitelist case.

6.3.2 Comparison Between Simulation Results of case 3 [LL-U-E] and case 5 [RR-CAP-E] (Blacklist case)

Figure 6.26 shows the required number of VMs needed to serve the cloud users in the attack mode (Blacklist case) while using the four mitigation technique. It is assumed that the blacklist table is periodically updated and it holds all the IP addresses of the attacking machines [7]. Consequently, mitigation technique (1) will drop the attack traffic directly while mitigation techniques (2), (3) and (4) will forward the attack requests to the V-node or VM Investigator for further verification. As a result, all the studied mitigation techniques will mitigate the attack traffic and will use only the initial running VMs in order to serve the legitimate users for all attack rates considered.

The response time results for the attack mode (blacklist case) after applying case 5 [RR-CAP-E] parameters are depicted in Figure 6.27. The response time for mitigation technique (1) is zero because all the legitimate users and attackers were blocked since their IP addresses already exist in the blacklist. In addition, the response time for mitigation techniques (2), (3) and (4) are constant as a result of blocking the attack traffic. Moreover, the response time for mitigation technique (2) is less than the response time for mitigation techniques (3) and (4). This is due to having some of the legitimate traffic in mitigation technique (2) being dropped due to having the same IP address and TTL value of a blacklisted user. Comparing Figure 6.28 to Figure 6.27 reveals a reduction in the cloud response time for case 3. The reason behind this reduction is related to the large relative difference between the average response time in solving CAPTCHA Turing test and the average response time in responding to a URL redirection as well as using LL algorithm instead of RR algorithm.

The resources utilization results for the attack mode (Blacklist case) after applying case 5 [RR-CAP-E] parameters are shown in Figure 6.29. Same trend for the resources utilization that is depicted in Figure 6.29 as the trend of the response time results shown in Figure 6.27.

Figure 6.30 shows the resources utilization results for the attack mode (Blacklist case) after applying case 3 [LL-U-E] parameters. A slight increase in the average cloud resources utilization is gained as a result of applying case 3 parameters to the four mitigation techniques. The reason behind this increase is due to the early finishing time of input traffic while using case 3 parameters as previously discussed in section 6.1.3.

Figure 6.31 shows the number of False Positive requests erroneously blocked by the EDoS mitigation techniques under study in the blacklist case. The entire legitimate traffic is blocked by mitigation technique (1) since legitimate users send requests toward the cloud service using IP addresses that are already blacklisted. Similar to the whitelist case that was explained in Figure 6.21 20% of the attack traffic have the same IP address and TTL value of legitimate users. Subsequently, mitigation technique (2) blocks the legitimate traffic that has the same IP address and TTL value of blacklisted users. On the other hand, mitigation technique (3) and (4) will successfully serve all the legitimate traffic since these mitigation techniques give another chance for the blacklisted users to proof their legitimacy through CAPTCHA test.

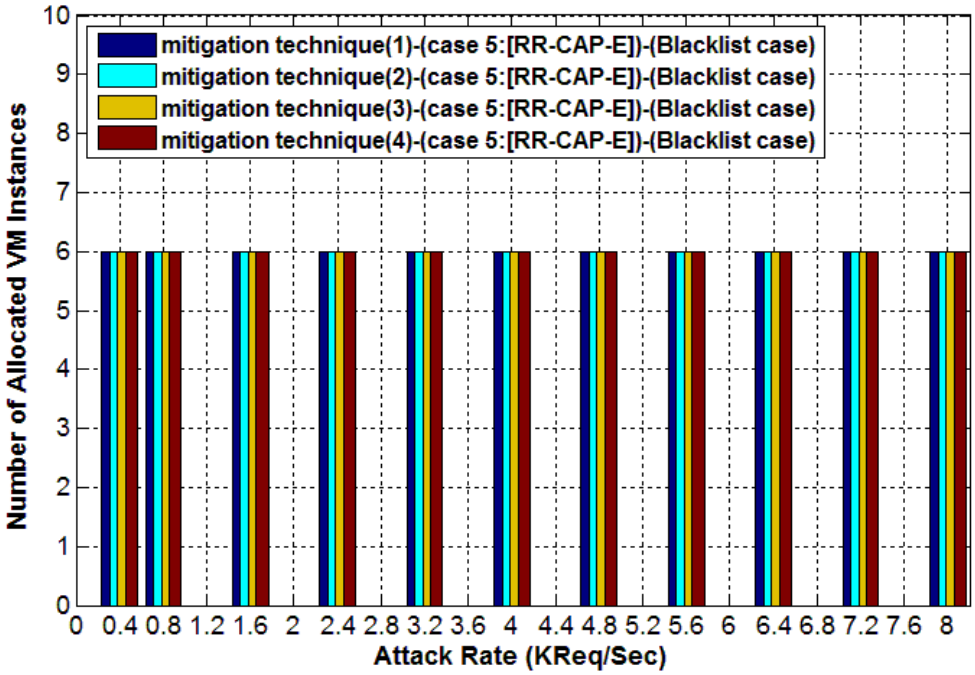


Figure 6.26: Simulation results of the number of allocated VMs at different attack rates for the Blacklist case.

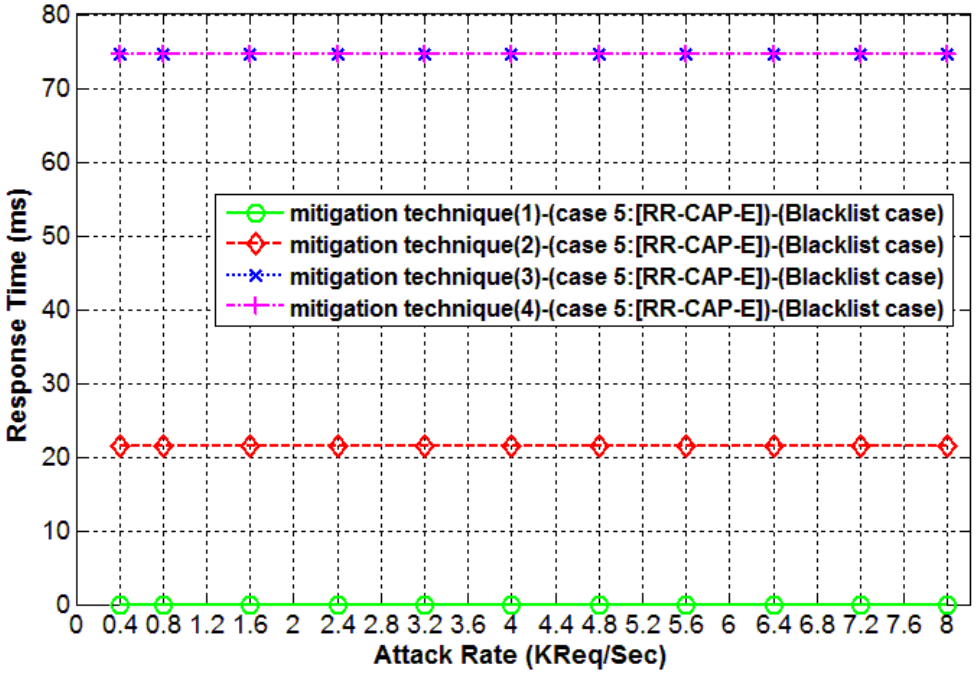


Figure 6.27: Response time results in the attack mode after applying case 5 [RR-CAP-E] for the Blacklist case.

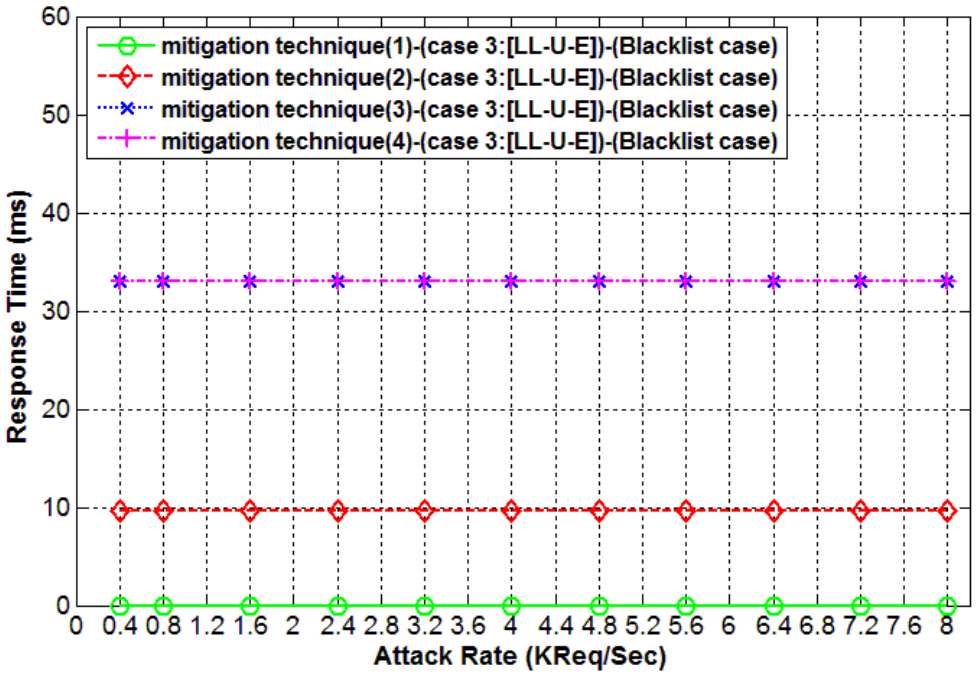


Figure 6.28: Response time results in the attack mode after applying case 3 [LL-U-E] for the Blacklist case.

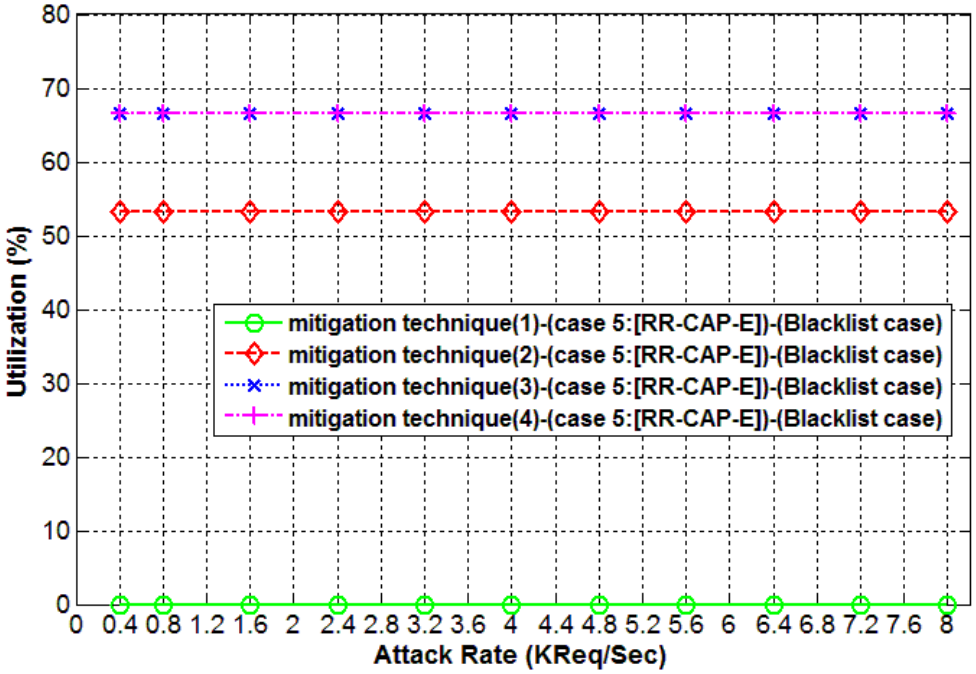


Figure 6.29: Utilization results in the attack mode after applying case 5 [RR-CAP-E] for the Blacklist case.

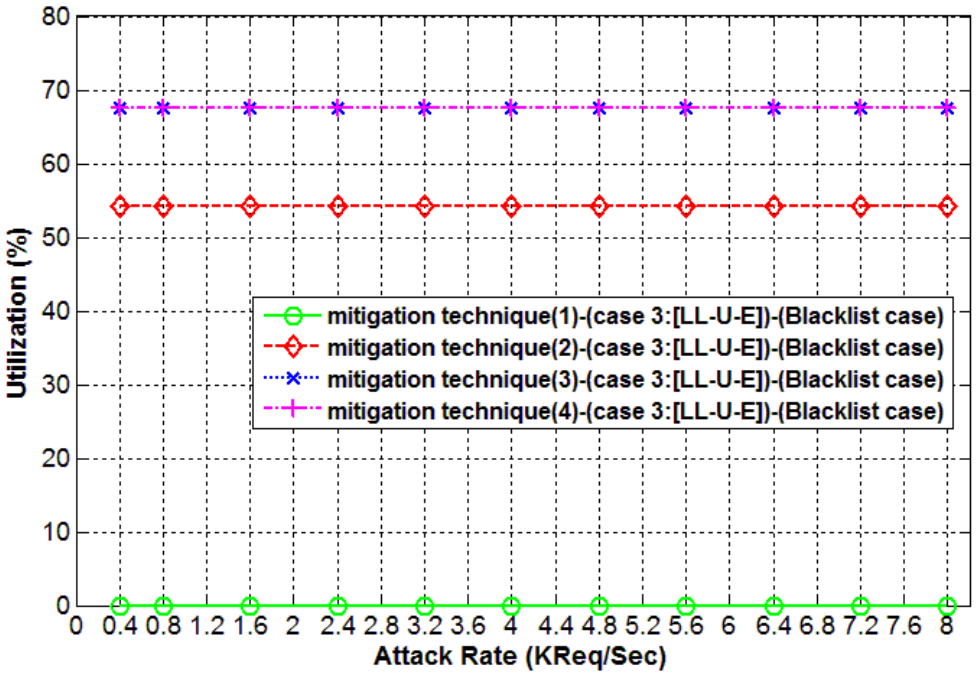


Figure 6.30: Utilization results in the attack mode after applying case 3 [LL-U-E] for the Blacklist case.

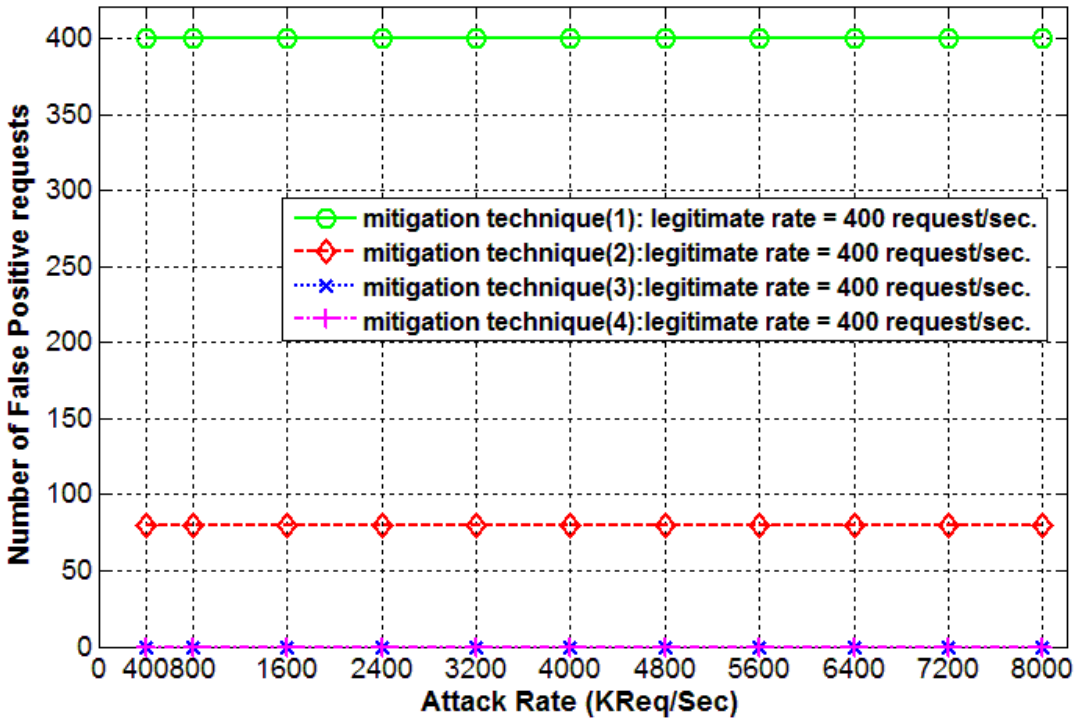


Figure 6.31: The number of False Positive requests in the attack mode for the blacklist case.

6.4 Attack Mode Results – Cloud Users Belong to The Same NAT-based Network

In this mode, the cloud resources are subjected to legitimate and attack traffic that are originating from behind the same NAT-based network. Thus, all the requests that are received by the cloud resources will share the same public IP address of the NAT router. This mode is one of the most possible yet most sophisticated scenario since it is difficult for regular EDoS mitigation techniques to differentiate between legitimate users and attackers that share the same public IP address.

To properly cover all the possibilities in this mode, the studied mitigation techniques are simulated under two cases: The whitelist case and the blacklist case. In both cases it is assumed that the public IP address is not used earlier to access the cloud service. In the whitelist case, initially an attacker that resides behind the NAT router targets the cloud resources with a legitimate request from the attacker's machine to place the NAT public IP address in the whitelist of the vFirewall. In the blacklist case, initially a bot machine behind a NAT router that is controlled by an attacker targets the cloud service, and subsequently the NAT public IP address will be placed in vFirewall blacklist table.

In order to evaluate the aforementioned cases for the mitigation techniques under study, the cloud service is subjected to a fixed load of legitimate traffic that equals 400 Req./Sec. and to an attack traffic that varies between 400 and 8000 Req./Sec. For every attack rate, the number of initial running instances in the studied mitigation technique is set to 6 VMs as deduced from the results shown in Figure 6.15.

It should be pointed out that to protect the internal structure of a private network behind the NAT, typically network administrators reset the TTL value of all outgoing traffic to the default value [55]. As such, it makes it harder for mitigation techniques to depend on the TTL value to distinguish between legitimate users and attackers from behind the NAT.

6.4.1 Comparison Between Simulation Results of case 3 [LL-U-E] and case 5 [RR-CAP-E] (Whitelist case)

The required number of VMs needed to serve the cloud users in the attack mode (whitelist case) after applying case 5 [RR-CAP-E] to the four mitigation techniques is shown in Figure 6.32. Both mitigation techniques (1) and (2) have allocated more VMs instances than mitigation techniques (3) and (4) because they consider the attack traffic as a flash traffic. In this case, all the attack traffic will bypass mitigation techniques (1) and (2) since the public NAT IP address is added to the vFirewall whitelist. Mitigation techniques (3) and (4) need to auto-scale one time in order to service cloud users as previously discussed in section 6.3.1.

Figure 6.33 shows the response time results for the attack mode (whitelist case) after applying case 5 [RR-CAP-E] parameters. Because all the attack traffic is considered as a flash traffic in mitigation technique (1) and (2), the response time increases whenever there is an increase in the attack rate. On the other hand, mitigation techniques (3) and (4) will eliminate the EDoS attack traffic when they are in operation but at the expense of relatively high response time. Figure 6.34 shows the response time for case 3 [LL-U-E] and it shows that there is a reduction in the cloud response time as compared to Figure 6.31. This reduction is a result of using the URL redirection instead of the CAPTCHA Turing test as a method to distinguish between legitimate clients and bot machines as well as using LL algorithm instead of RR algorithm.

The resource utilization results in the attack mode (whitelist case) after applying case 5 [RR-CAP-E] parameters are depicted in Figure 6.35. The CPU utilization results for mitigation (1) and (2) are similar because both techniques use the same number of VMs to serve the cloud users. Also, the CPU utilization results for mitigation (3) and (4) are similar for the same reason. Comparing Figure 6.35 to Figure 6.36 reveals a slight increase in the average cloud resources utilization for case 3 due to the early finishing of input traffic after applying case 3 parameters as previously discussed in section 6.1.3.

Figure 6.37 shows the number of False Negative requests served by the cloud resources for the whitelist case. The entire attack traffic is severed by the cloud resources in mitigation techniques (1) and (2) since the public NAT IP address is added to the vFirewall whitelist. On the other hand, mitigation techniques (3) and (4) will serve some of the attack traffic until the utilization threshold is crossed as previously noted when explaining Figure 6.25.

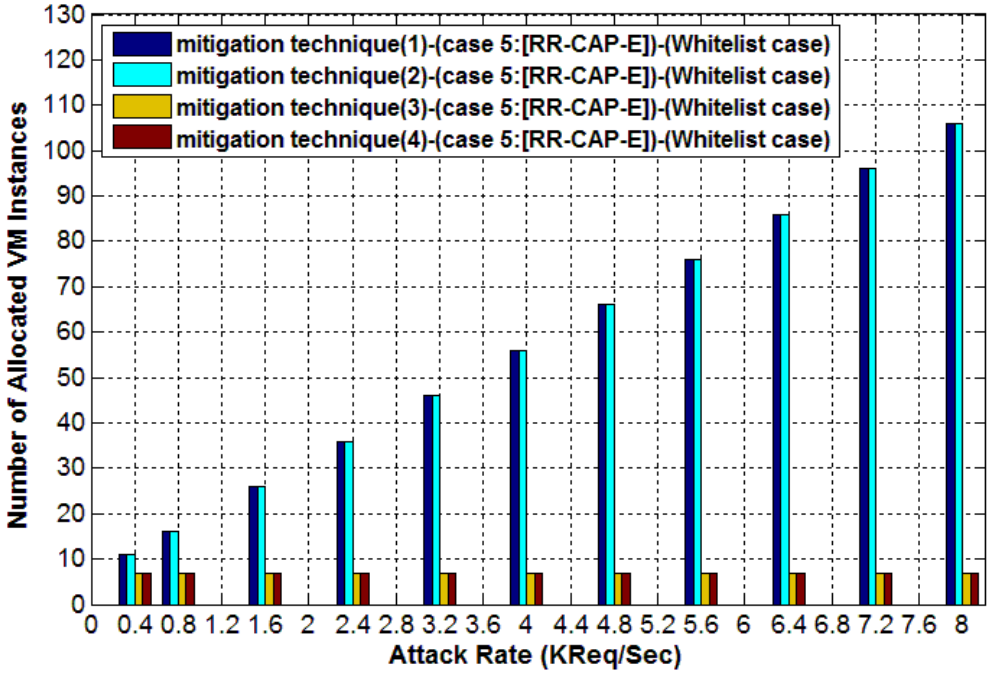


Figure 6.32: Simulation results of the number of allocated VMs at different attack rates for the Whitelist case after applying case 5 [RR-CAP-E].

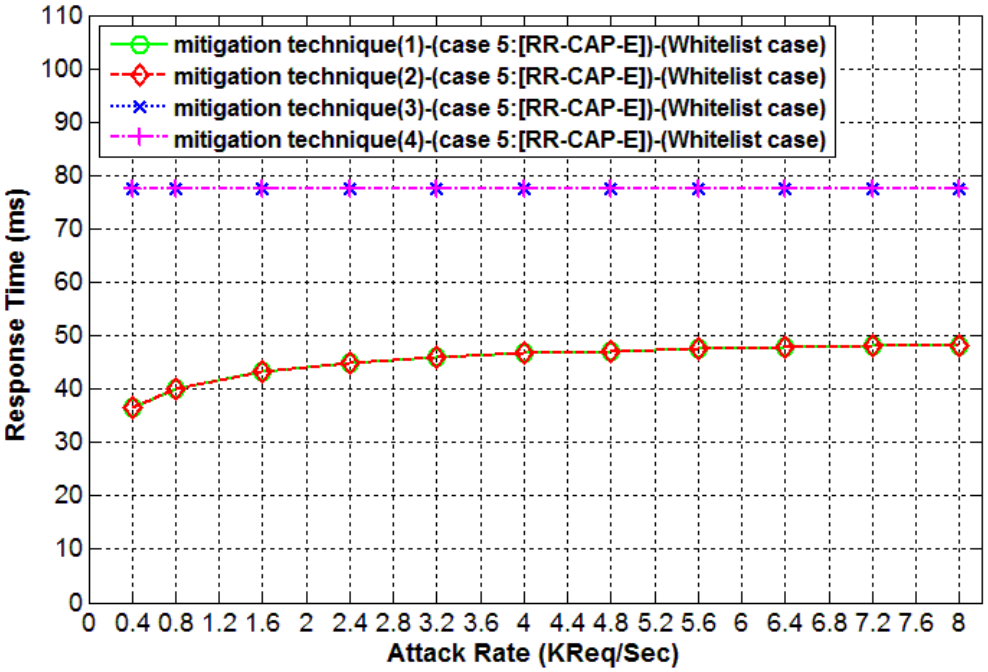


Figure 6.33: Response time results in the attack mode after applying case 5 [RR-CAP-E] for the whitelist case.

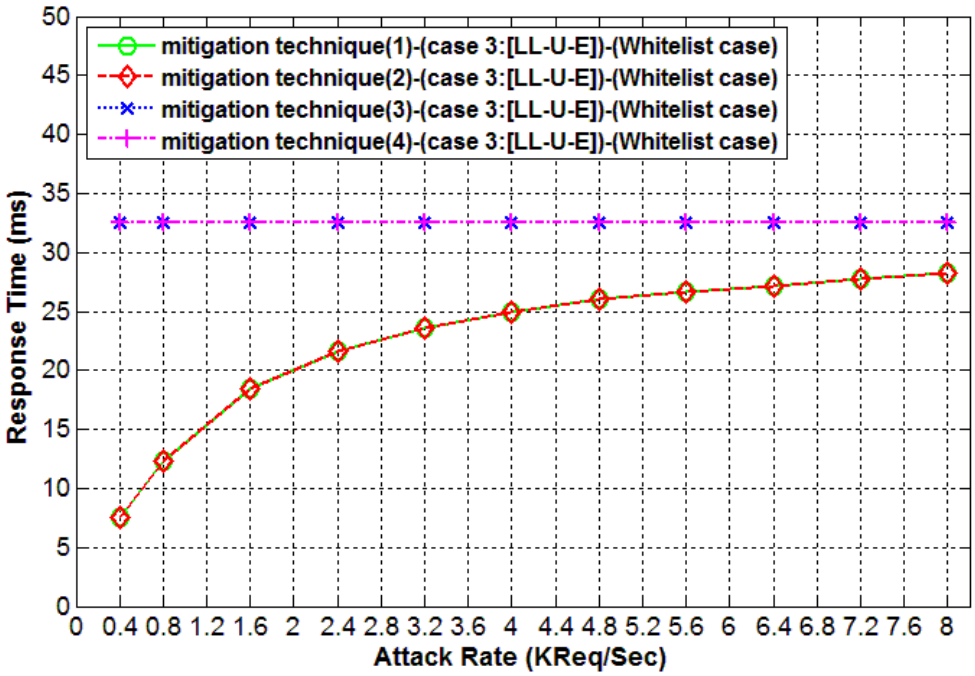


Figure 6.34: Response time results in the attack mode after applying case 3 [LL-U-E] for the whitelist case.

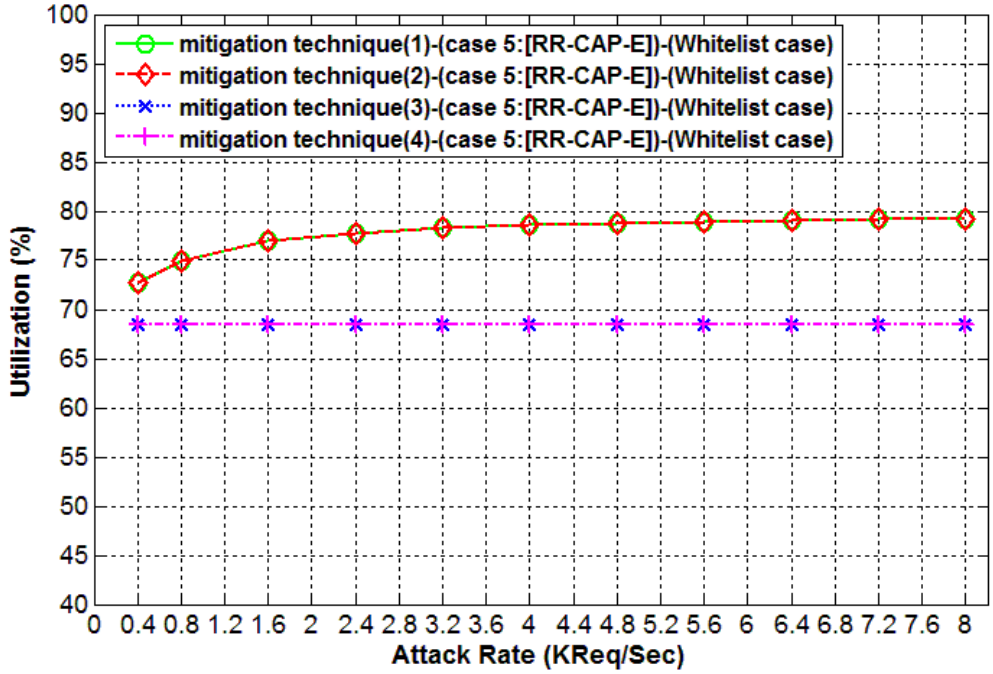


Figure 6.35: Utilization results in the attack mode after applying case 5 [RR-CAP-E] for the whitelist case.

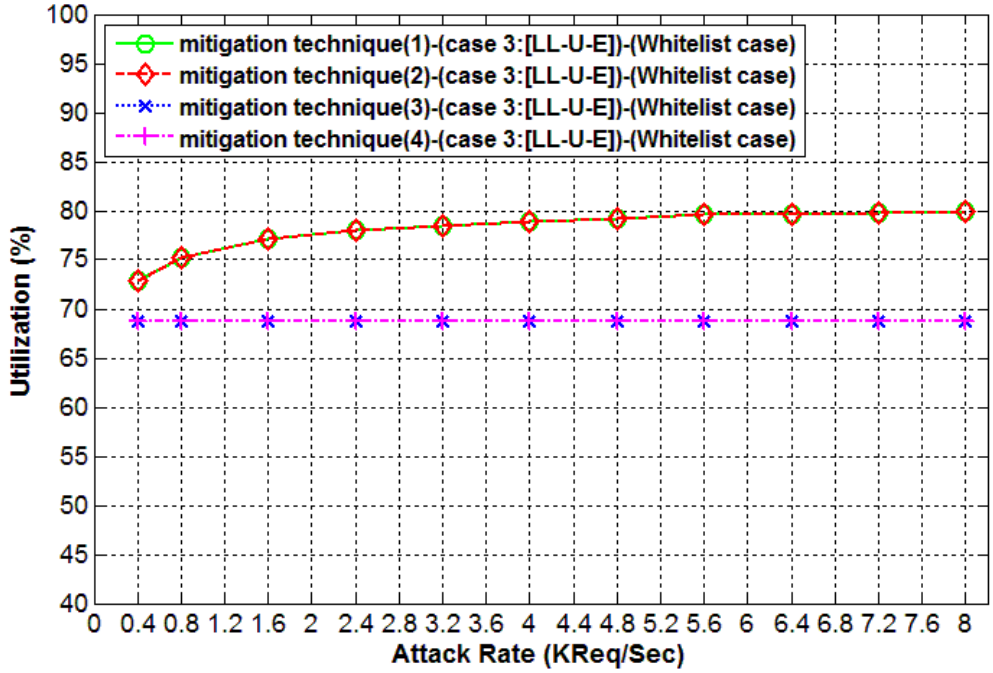


Figure 6.36: Utilization results in the attack mode after applying case 3 [LL-U-E] for the whitelist case.

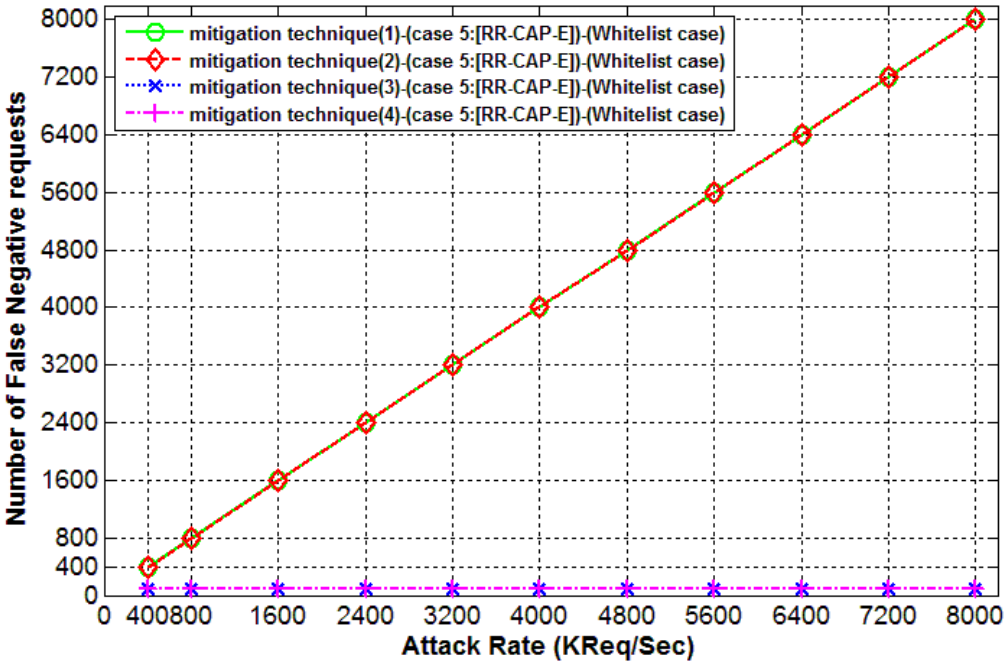


Figure 6.37: The number of False Negative requests in the attack mode for the whitelist case.

6.4.2 Comparing Between Simulation Results of case 3 [LL-U-E] and case 5 [RR-CAP-E] (Blacklist case)

The required number of VMs needed to serve the cloud users in the attack mode (blacklist case) after applying case 5 [RR-CAP-E] to the four mitigation techniques is shown in Figure 6.38. It is assumed that the blacklist table is periodically updated and it holds the public IP of the NAT router. Consequently, mitigation techniques (1) and (2) will block the entire NAT. On the other hand, mitigation techniques (3) and (4) will direct all the traffic that comes from behind the NAT router to the VM investigator for further analysis. Accordingly, none of the four mitigation techniques need to perform auto scaling and they will operate using the initial running instances.

Figure 6.39 shows the response time results for the attack mode (blacklist case) after applying case 5 [RR-CAP-E] parameters. In mitigation techniques (1) and (2) the response time is zero because all the legitimate users and attackers were blocked since their IP addresses already exist in the blacklist. In addition, the response time in mitigation techniques (3) and (4) is constant as an evidence of blocking the attack traffic but with the expense of a relatively high response time. This is because all the legitimate users need to solve CAPTCHA Turing test in order to gain access to the cloud resources. Figure 6.40 shows a reduction in the cloud response time for case 3 when compared with Figure 6.39. This reduction is a result of using the URL redirection instead of the CAPTCHA Turing test as a method to distinguish legitimate clients from bot machines as well as using LL algorithm instead of RR algorithm.

The resources utilization results for the attack mode (Blacklist case) after applying case 5 [RR-CAP-E] parameters are shown in Figure 6.41. Same trend for the resources utilization

that is depicted in Figure 6.41 as the trend of to the response time results shown in Figure 6.39.

Figure 6.42 shows the resources utilization results for the attack mode (Blacklist case) after applying case 3 [LL-U-E] parameters. A slight increase in the average cloud resources utilization is gained as a result of applying case 3 parameters to the four mitigation techniques. The reason behind this increase is due to the early finishing time of input traffic while using case 3 parameters as previously discussed in section 6.1.3.

Figure 6.43 shows the number of False Positive requests erroneously blocked by the EDoS mitigation techniques under study in the blacklist case. Note that Figure 6.43 assumes that there are 400 legitimate Req./Sec. being received for all attack mitigation technique. The entire legitimate traffic is blocked by mitigation techniques (1) and (2) since the NAT public IP address is added to the vFirewall blacklist. On the other hand, mitigation techniques (3) and (4) will successfully serve all the legitimate traffic since these mitigation techniques give another chance for the blacklisted users to proof their legitimacy through CAPTCHA test.

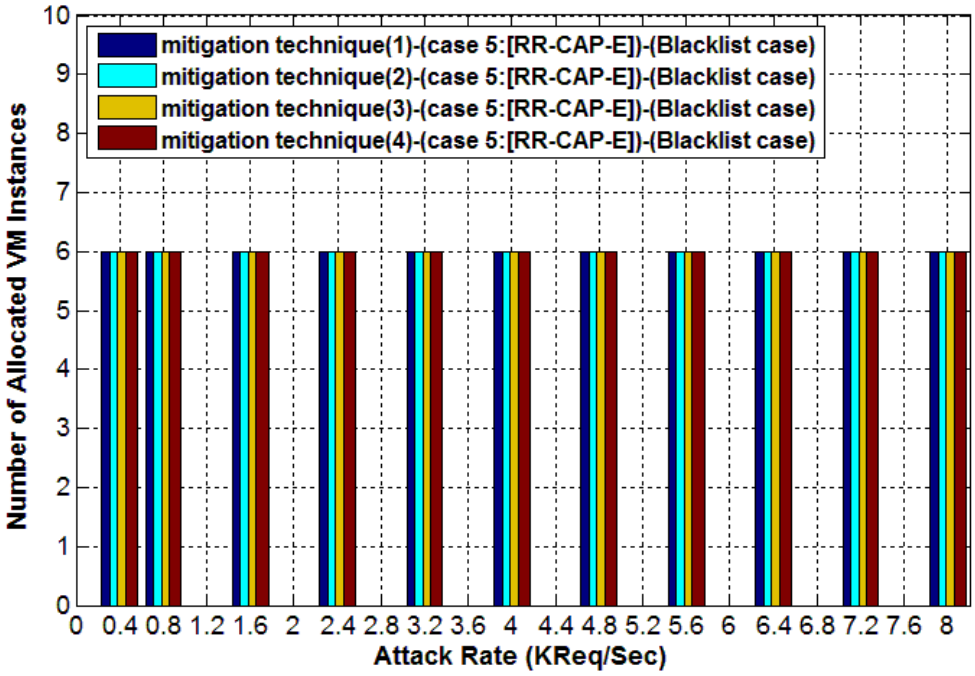


Figure 6.38: simulation results of the number of allocated VMs at different attack rates for the Blacklist case after applying case 5 [RR-CAP-E].

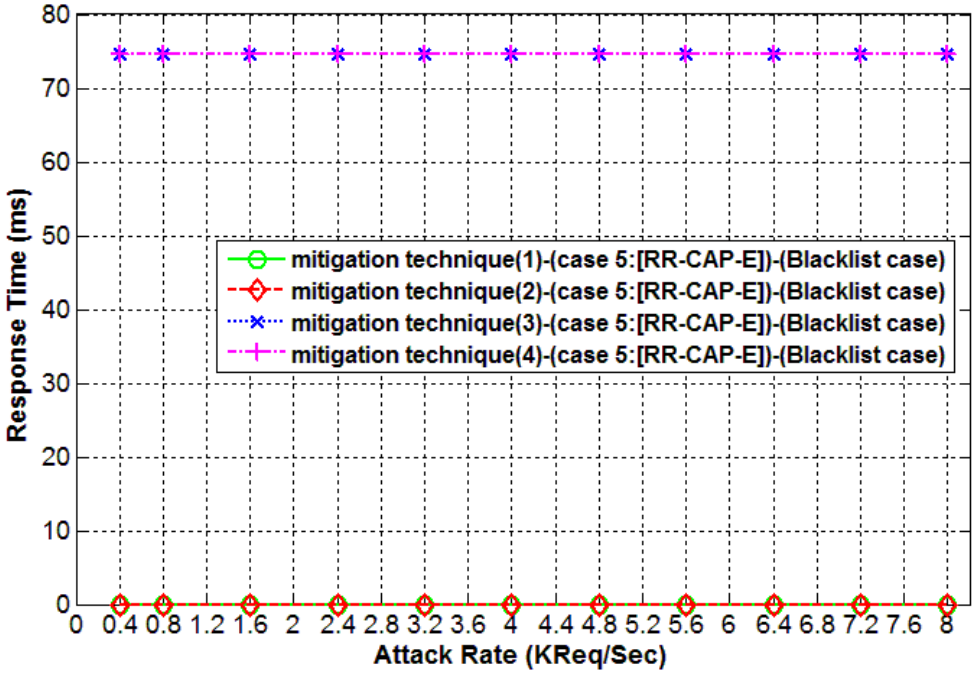


Figure 6.39: Response time results in the attack mode after applying case 5 [RR-CAP-E] for the Blacklist case.

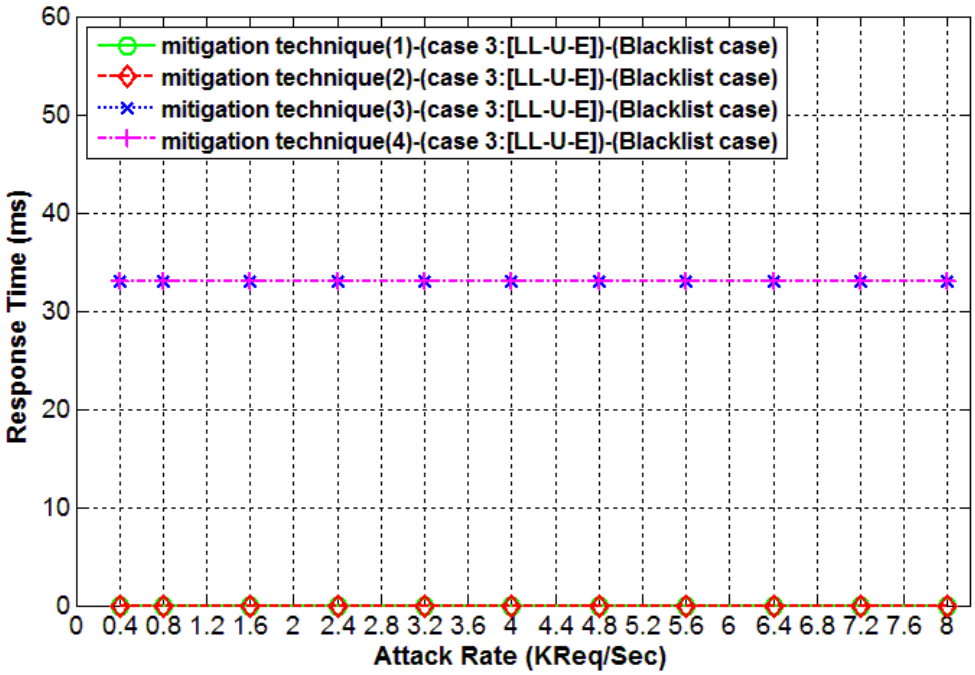


Figure 6.40: Response time results in the attack mode after applying case 3 [LL-U-E] for the Blacklist case.

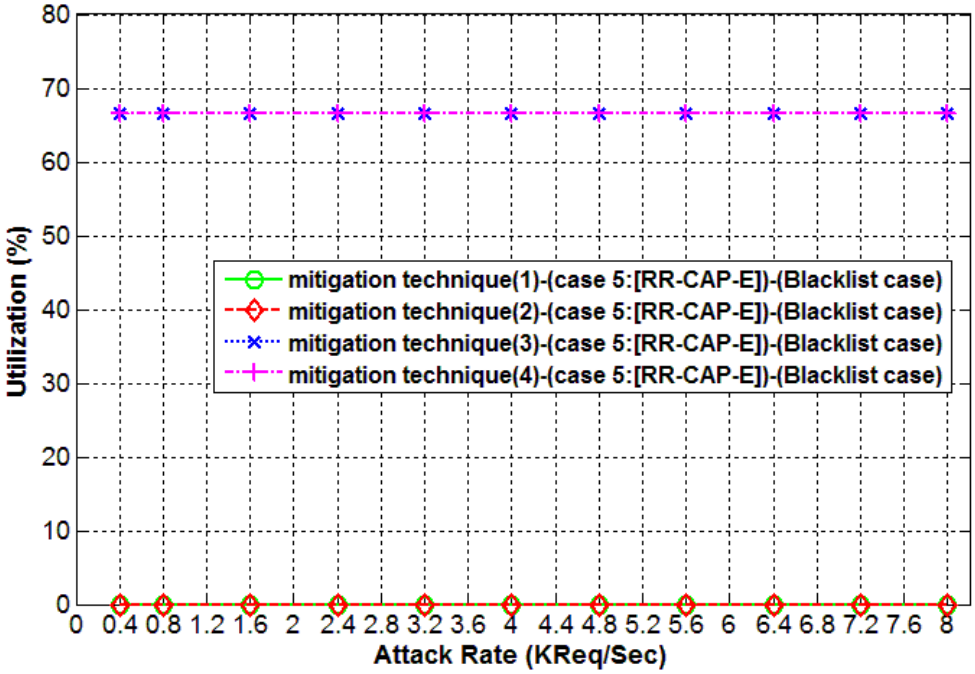


Figure 6.41: Utilization results in the attack mode after applying case 5 [RR-CAP-E] for the Blacklist case.

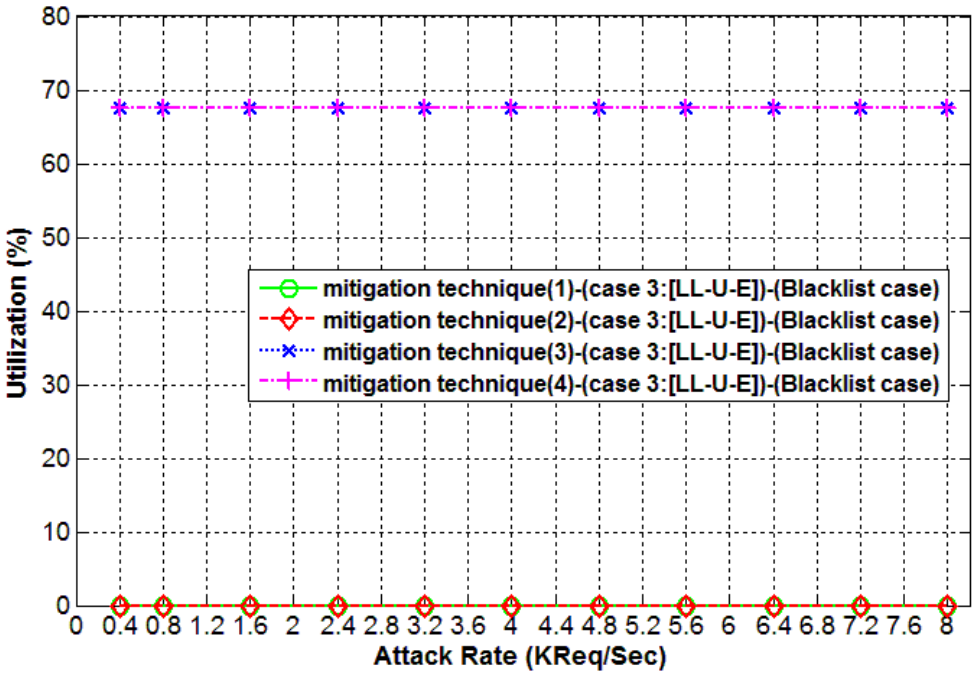


Figure 6.42: Utilization results in the attack mode after applying case 3 [LL-U-E] for the Blacklist case.

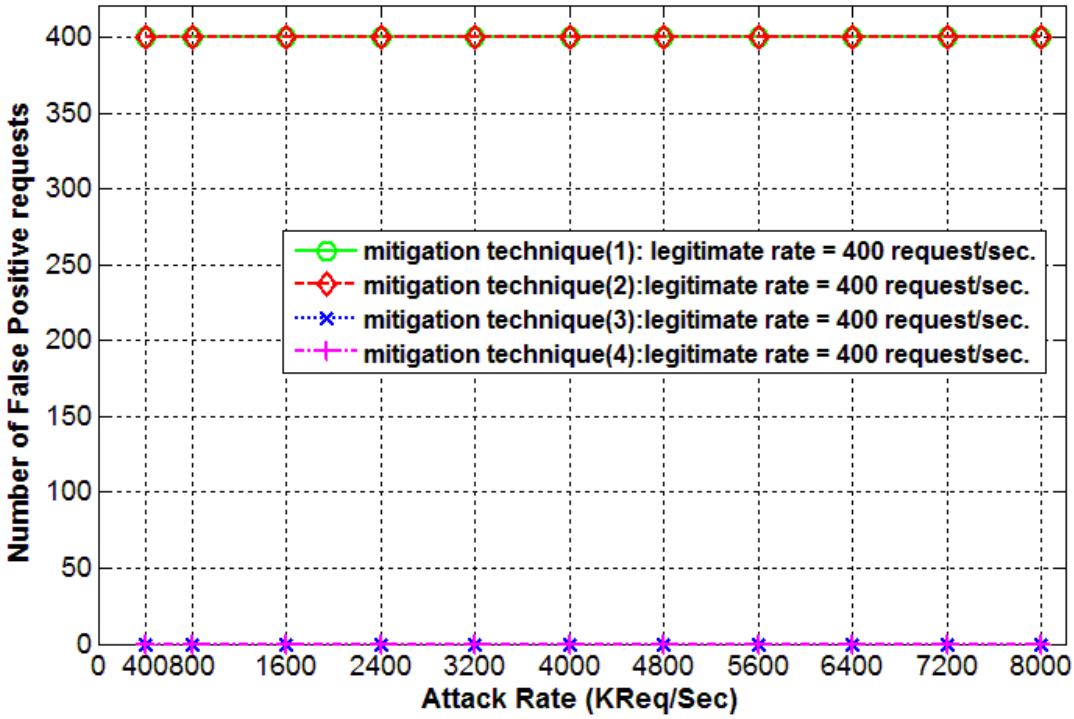


Figure 6.43: The number of False Positive requests in the attack mode for the blacklist case.

CHAPTER 7 CONCLUSION AND FUTURE WORK

This chapter summarizes the major contributions and findings in this thesis. It also provides a list of possible future research directions.

7.1 Conclusion

Many researchers proposed mitigation techniques that can reduce the effect of the EDoS attack. However, the literature lacks a concrete study that can help cloud service providers to choose between the different alternatives. Thus, a common platform to evaluate any EDoS mitigation technique was developed and proposed by this thesis. A performance evaluation between four of the existing mitigation techniques was conducted. Such a performance evaluation platform can be easily changed to test the performance of future solutions. As a result of the evaluation process of the studied EDoS mitigation techniques, it is advisable to include some features in any future EDoS mitigation technique. The features include the use of the LL algorithm instead of the RR algorithm as the cloud load scheduler. Another feature that can be included is the use of the URL redirection technique to identify automated attackers especially when they belong to a NAT-based network. Finally, it is advisable to test the EDoS mitigation technique under different modes including its adaptability to an EDoS attack, the flash overcrowd phenomena, and the IP spoofing problem.

Another conclusion from this thesis is the need to test the performance of the mitigation technique using different probability distributions for request service times so as to model the real life cloud environment as much as possible.

7.2 Future work

The future work improvements will look into the following aspects:

1- Testing the mitigation technique under other real cloud conditions such as changing the VMs capacity, using different load balancing technique other than RR and LL, and using different probability distribution other than exponential and Pareto to characterize the arrival process and request service times.

2- Study other queuing system models for modeling the VM instances in the cloud including a parallel Pareto/Pareto/1 queuing system [52], and a parallel G/M/1 queuing system with three alternatives for interarrival time distributions: Hypergeometric, Exponential, and Pareto distributions [51].

3- Study other performance metrics including the throughput of legitimate clients, and the cost associated with each mode while considering different pricing models.

4- Check the suitability of the studied mitigation techniques for reducing the effect of the fraudulent resource consumption attack or low rate EDoS attack (LR-EDoS).

5- Consider cloud optimization problems for tuning some of the provisioning mechanism parameters such as the lower utilization threshold that is used for the auto scaling.

6- Evaluate the EDoS mitigation techniques when the attackers are capable of solving CAPTCHA Turing tests by utilizing some automatic software solvers like Xrumer [56].

References

[1] P. Mell, and T. Grance, "The NIST Definition of Cloud Computing," URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, 2011.

[2] S. Khor, and A. Nakao, "sPoW: On-demand cloud-based eddos mitigation mechanism," in *Proc. of the Fifth Workshop on Hot Topics in System Dependability,* 2009.

[3] M. Kumar, P. Sujatha, and P. Dharanyadevi, "Mitigating EDoS in Cloud Computing using In-Cloud Scrubber Service: A detailed study with Novel Approach for Extenuating DDoS in Cloud Computing," Lap LAMBERT Academic Publishing, 2012.

[4] M. Saleh, and A. Azizah, "A Novel Protective Framework for Defeating HTTP-Based Denial of Service and Distributed Denial of Service Attacks," *The Scientific World Journal,* 2014.

[5] M. H. Sqalli, F. Al-Haidari, and K. Salah, "EDoS-Shield-A Two-Steps Mitigation Technique against EDoS Attacks in Cloud Computing," *Fourth IEEE International Conference on Utility and Cloud Computing*, 2011.

[6] F. Al-Haidari, M. H. Sqalli, and K. Salah, "Enhanced Edos-Shield for mitigating Edos attacks originating from spoofed IP addresses," in *the Security and Privacy in Computing and Communications (TrustCom),* 2012.

[7] M. Kumar, R. Korra, P. Sujatha, and Mu. Kumar, "Mitigation of economic distributed denial of sustainability (EDDoS) in cloud computing," in *Proc. of the International Conference on Advances in Engineering and Technology,* 2011.

[8] M. Kumar, P. Sujatha, V. Kalva, R. Nagori, A. Katukojwala, and Mu. Kumar, "Mitigating economic denial of sustainability (edos) in cloud computing using in-cloud scrubber service," *in the Fourth International Conference on Computational Intelligence and Communication Networks,* 2012.

[9] S. V. Sandar, and S. Shenai, "Economic Denial of Sustainability (EDoS) in Cloud Services using HTTP and XML based DDoS Attacks," *International Journal of Computer Applications, Vol. 41, No. 20, pp. 11-16,* 2012.

[10] M. Masood, Z. Anwar, S. Raza, and M. Hur, "EDoS Armor: A cost effective economic denial of sustainability attack mitigation framework for e-commerce

applications in cloud environments," in *Multi Topic Conference (INMIC), 2013 16th International,* 2013.

[11] W. Alosaimi, and K. Al-Begain, "A New Method to Mitigate the Impacts of the Economical Denial of Sustainability Attacks Against the Cloud," in *Proceedings of the 14th Annual Post Graduates Symposium on the convergence of Telecommunication, Networking and Broadcasting (PGNet), pp. 116-121. Liverpool- UK: Liverpool John Moores University,* 2013.

[12] W. Alosaimi, and K. Al-Begain, "An Enhanced Economical Denial of Sustainability Mitigation System for the Cloud," in *Seventh International Conference on Next Generation Mobile Apps, Services and Technologies (NGMAST),* 2013.

[13] M. Kumar, and N. Roberts, "A technique to reduce the economic denial of sustainability (EDoS) attack in cloud," in *2013 Advanced Research in Engineering and Technology, volume 2, pp. 571 – 574,* 2013.

[14] Z. A. Baig, and F. Binbeshr, "Controlled Virtual Resource Access to Mitigate Economic Denial of Sustainability (EDoS) Attacks against Cloud Infrastructures," in *the International Conference on Cloud Computing and Big Data (CloudCom-Asia),* 2013.

[15] Amazon Web Service Incorporation, "Amazon Cloud Watch: monitoring for AWS cloud resources," 2013, available at http://aws.amazon.com/cloudwatch/ .

[16] P. Singh, S. Manickam, and S. Ul-Rehman, "A survey of mitigation techniques against Economic Denial of Sustainability (EDoS) attack on cloud computing architecture," *International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions),* 2014.

[17] R. Thaper, and A. Verma, "A Survey on Economic Denial of Sustainability Attack Mitigation Techniques," *International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 3, March* 2015.

[18] K. Anusha, N. Tulasiram, and S. Mary, "Detection of Economic Denial of Sustainability Using Time Spent on a Web Page in Cloud," in *IEEE International Conference on Cloud Computing in Emerging Markets (CCEM),* 2013.

[19] B. Saini, and G. Somani, "Index Page Based EDoS Attacks in Infrastructure Cloud," in *Proc. SNDS, pp.382-395,* 2014.

[20] F. Al-Haidari, M. H. Sqalli, and K. Salah, "Evaluation of the Impact of EDoS Attacks Against Cloud Computing Services," *Arabian Journal for Science and Engineering, Volume 40, Issue 3, pp 773-785,* 2014.

[21] Z. A. Baig, S. M. Sait, and F. Binbeshr, "Controlled Access to Cloud Resources for Mitigating Economic Denial of Sustainability (EDoS) Attacks," *Computer Networks,* 2016.

[22] M. Ficco, and M. Rak, "Economic Denial of Sustainability Mitigation in Cloud Computing," *Organizational Innovation and Change. Springer International Publishing*, p229-238, 2016.

[23] M. Mehra, M. Agarwal, R. Pawar, and D. Shah, "Mitigating denial of service attack using CAPTCHA mechanism," *ICWET Workshop on Emerging Trends in Technology, ACM New York, NY, USA*, pp. 284-287, 2011.

[24] L. Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "reCAPTCHA: Human-Based Character Recognition via Web Security Measures," *Science*, pp. 1465-1468, Sep. 2008.

[25] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities," in *High Performance Computing & Simulation (HPCS'09) International Conference on IEEE*, 2009.

[26] R. N. Calheiros et al., "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience 41.1 (2011): 23-50.

[27] N. Singh, S. P. Ghrera, and P. Chaudhuri, "Denial of Service Attack: Analysis of Network Traffic Anormaly using Queuing Theory," *Journal of Computer Science and Engineering, Vol. 1, Issue 1, pp. 48-54*, May 2010.

[28] Y. Wang, C. Lin, Q. Li, Y. Fang, "A Queueing Analysis for the Denial of Service (DoS) Attacks," *Computer Networks 51, 3564–3573*, 2007.

[29] K. Chandy and C. Sauer, "Approximate methods for analyzing queueing network models of computing systems," *Journal of ACM Computing Surveys, vol. 10, no. 3, pp. 281-317*, Sep. 1978.

[30] L. Kleinrock, "Queueing Systems: Theory," vol. 1, New York, Wiley, 1975.

[31] D. Gross, J. F. Shortle, J. M. Thompson, and C. M. Harris, "Fundamentals of Queuing Theory," John Wiley and Sons Inc., 2008.

[32] K. C. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," in *the Proceedings of INET 1998, Geneva, Switzerland*, July 1998.

[33] D. Bellenger, et al., "Scaling in cloud environments" *Recent Researches in Computer Science 33,* 2011.

[34] K. Salah, K. Elbadawi, and R. Boutaba, "Performance modeling and analysis of network firewalls," *IEEE Transactions on Network and Service Management, vol. 9, no. 1, pp. 12–21, 2012.*

[35] H. Liu, and S. Wee, "Web Server Farm in the Cloud: Performance Evaluation and Dynamic Architecture," in *Proc. of the 1st International Conference on Cloud Computing (CloudCom 2009),* Dec 2009.

[36] A. Wool, "A quantitative study of firewall configuration errors," *IEEE Computer*, 37(6), 2004.

[37] S. Islam, K. Lee, A. Fekete, and A. Liu, "How A Consumer Can Measure Elasticity for Cloud Platforms," technical report, SCHOOL OF INFORMATION TECH, Univercity of Sydeny, 2011.

[38] Amazon EC2 Pricing, available from http://aws.amazon.com/ec2/pricing/

[39] H. Wang, C. Jin, and K. Shin, "Defense against Spoofed IP Traffic Using Hop-Count Filtering," *IEEE/ACM Trans. Networking, vol.15 No. I, Feb. 2007.*

[40] D. Moore, G. Voelker, and S. Savage, "Inferring Internet Denial-of-Service Activity," *ACM Transactions on Computer Systems (TOCS), USA, vol. 24, Issue 2, pp. 115 – 139, May 2006.*

[41] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, M. A. Kozuch, "Towards Understanding Heterogeneous Clouds At Scale: Google Trace Analysis," Technical Report, Intel Science and Technology Center for Cloud Computing, 2012.

[42] Y. M. Teo, R. Ayani, "Comparison of load balancing strategies on cluster-based web servers," *Simulation (San Diego, Calif.), 77.05-6: 185-195*, 2001.

[43] H. S. Cho, and Y. K. Noh, "System for preventing normal user being blocked in network address translation (NAT) based web service and method for controlling the same," U.S. Patent No 8,434,141, 2013.

[44] Beware HTTP Redirects for SEO and Performance. Available on: http://www.websiteoptimization.com/speed/tweak/redirect/

[45] Y. Lee, and C. Hsu, "Usability study of text-based CAPTCHAs," *Displays, 32.2: 81-86*, 2011.

[46] X. Nan, Y. He, and L. Guan, "Optimal resource allocation for multimedia cloud based on queuing model," in *IEEE 13th International Workshop on Multimedia Signal Processing*, 2011.

[47] R. N. Calheiros, R. Ranjan, and R. Buyya, "Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments," in *International Conference on Parallel Processing*, 2011.

[48] R. Pal, and P. Hui, "Economic models for cloud service markets," in *International Conference on Distributed Computing and Networking, Springer Berlin Heidelberg, p. 382-396*, 2012.

[49] Y. Shi, X. Jiang, and K. Ye, "An energy-efficient scheme for cloud resource provisioning based on CloudSim," in *IEEE International Conference on Cluster Computing*, 2011.

[50] A. Downey, "Lognormal and Pareto distributions in the Internet," *Computer Communications, vol.28, no.7, pp.790-801*, 2005.

[51] J. Gordon, "Pareto process as a model of self-similar packet traffic," *Global Telecommunications Conference, GLOBECOM'95, IEEE. Vol. 3. IEEE*, 1995.

[52] A. G. Fayoumi, "Performance evaluation of a cloud based load balancer severing Pareto traffic," *Journal of Theoretical and Applied Information Technology 32.1, 28-34, 2011*.

[53] C. Sutton, and M. I. Jordan, "Bayesian inference for queueing networks and modeling of internet services," *The Annals of Applied Statistics, 254-282,* 2011.

[54] W. Dawoud, I. Takouna, and C. Meinel., "Elastic VM for rapid and optimum virtualized resources allocation," *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on. IEEE,* 2011.

[55] Y. Gokcen, V. A. Foroushani, and A. NurZincir-Heywood, "Can we identify NAT behavior by analyzing Traffic Flows?," *Security and Privacy Workshops (SPW), IEEE,* 2014.

[56] M. Motoyama, et al. "Re: CAPTCHAs-Understanding CAPTCHA-Solving Services in an Economic Context," in *USENIX Security Symposium. p. 3.* 2010.

APPENDIX: CLOUDSIM SIMULATOR CODE DESIGN

The main six CloudSim simulator Modules that are related to the operation of the four EDoS mitigation techniques under study are shown in Figure A.1. The six Modules are the Main Module, the DataBase Module, the Cloudlet (CL) Module, the Broker Module, the Cloud Information service (CIS) Module, and the DataCenter Module. The Main Module and the DataBase Module are needed for the operation of the other modules. Figure A.1 provides the additional features added to the CloudSim simulator for building the considered EDoS mitigation techniques. The additional features are shown using underlined text in Figure A.1.

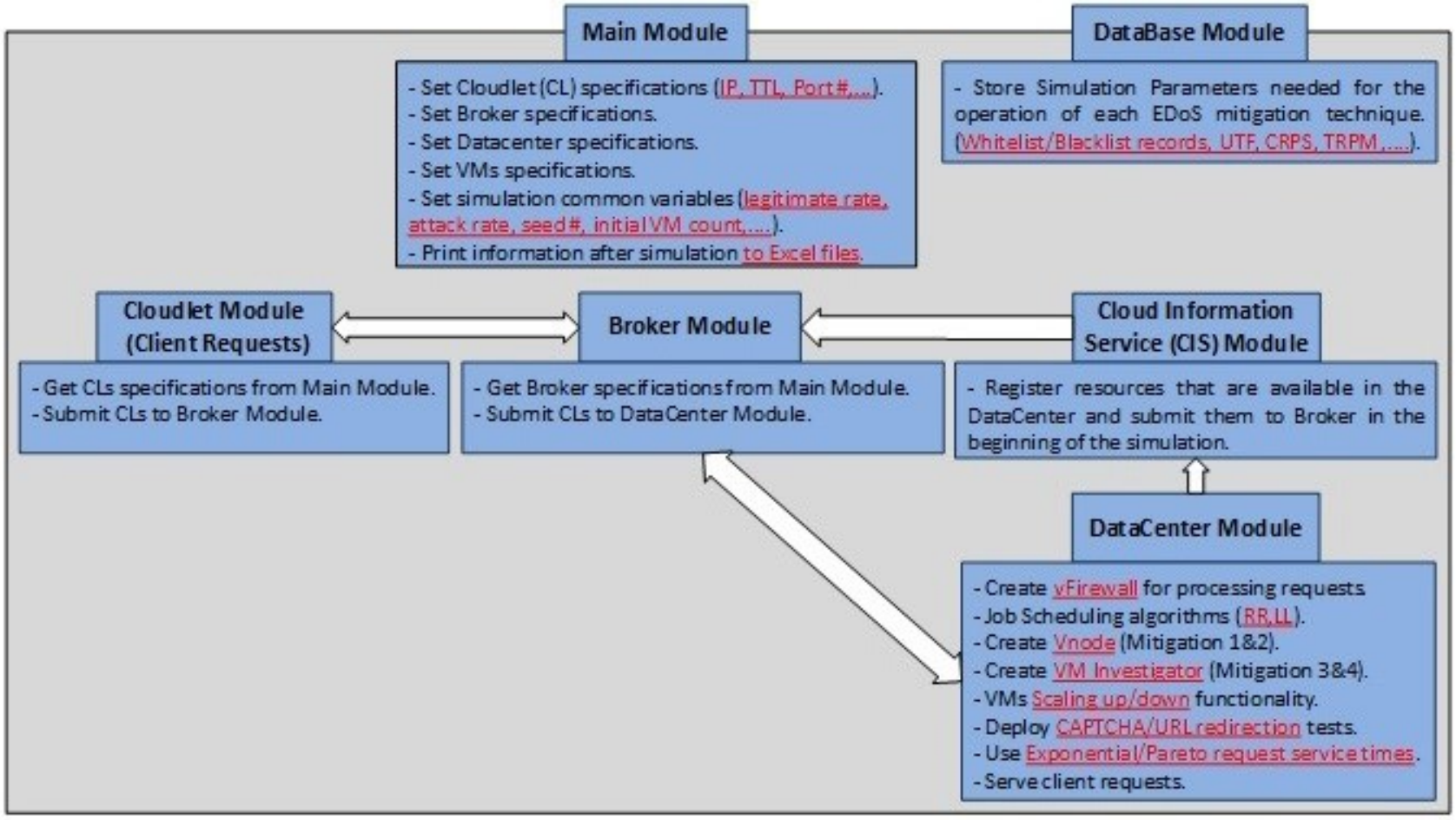


Figure A.1: Main functionalities of the CloudSim and the modified features.

Vitae

Name : Omar Ali Adel Maraqa

Nationality : Palestinian

Date of Birth : 5/13/1988

Email : Eng.maraqa@hotmail.com

Address : Hebron-Palestine

Academic Background : Complete M.S. in Computer Networks from Computer Engineering department at King Fahd University of Petroleum and Minerals (KFUPM) at Dec 2016 , earned the B.S. degree in Communication and Electronics Engineering from Palestine Polytechnic University Hebron-Palestine 2011. Research interests include Energy Efficient Cellular Networks, Cognitive Radio and Security concerns in Cloud Computing.